



Universidad
Carlos III de Madrid

GRADO EN INGENIERÍA INDUSTRIAL
Y AUTOMÁTICA

TRABAJO DE FIN DE GRADO

Diseño y construcción de una mascota doméstica

Autor: Miguel Ángel JUZGADO CASCO

Tutor: LUIS ENRIQUE MORENO LORENTE

Marzo 2016





Agradecimientos:

Me gustaría dar cabida a unas pocas líneas de agradecimiento en el presente documento.

A mi familia, por el apoyo prestado.

En especial a mis padres, por darme la oportunidad de llegar hasta aquí.

A mi hermana, por las ayudas prestadas hasta hoy.

A mi pareja, por demostrarme que siempre está ahí cuando la necesito y por la paciencia que ha tenido conmigo durante el desarrollo del proyecto.

A mis amigos, que han estado presente todos estos años de estudios y que espero que estén el resto de la vida.

Por último a los profesores, empezando por mi profesora de Educación Infantil hasta el tutor del proyecto, pasando por los maestros del instituto y de la carrera cursada.

Sin todas estas personas detrás de mí, nada de esto hubiera sido posible.

GRACIAS.



RESUMEN

El presente proyecto consiste en el diseño y la construcción de una mascota doméstica. Ésta será un robot autónomo de pequeñas dimensiones, cuyas principales funciones serán la teleoperación y la visión por streaming de la cámara incorporada. Dicho trabajo ha sido realizado como Trabajo de Fin de Grado de la titulación de Grado en Ingeniería Electrónica Industrial y Automática de la Escuela Politécnica Superior de la Universidad Carlos III de Madrid.

La navegación del robot se podrá realizar de forma manual o de forma automática, gracias a la teleoperación y a los sensores incorporados respectivamente.

Una de las funciones del proyecto es la de proporcionar al usuario un total manejo del pequeño robot gracias a la teleoperación. La conexión entre el mando y el robot se realiza a través de Bluetooth y mediante ella, el usuario podrá mover al robot hacia delante, rotar para ambos lados y encender una pequeña linterna que lleva incorporada, por si el robot se introduce en un entorno oscuro o con poca luz.

El usuario será capaz de ver el camino que el robot va tomando gracias a la cámara que lleva incorporada. La reproducción del vídeo será a través de streaming, por lo que el usuario podría seguir los pasos del robot a través del monitor de un ordenador o de su propio smartphone.

La otra modalidad que permite el robot es la del Modo Automático, es decir, que el robot se guía solo mediante la luz recibida, es decir, que cuando el robot detecta una luz, se mueve hacia ella.

El robot ha sido fabricado en madera de contrachapado por su fácil manejo, su buena resistencia y su poco peso. También se ha implementado el sistema electrónico y se ha programado la placa controladora del robot, para que el robot pueda realizar todas las funciones descritas anteriormente.

Se puede decir que se ha conseguido realizar la construcción de un robot doméstico con varias funciones y que para la realización de dicho trabajo no hace falta un gran gasto económico. También que es un proyecto abierto a evolucionar, ya que en el futuro se le pueden añadir más aplicaciones o simplemente, cambiar las que tiene actualmente con una simple reprogramación.



Índice

RESUMEN.....	4
Índice de figuras	7
Índice de tablas.....	9
Capítulo 1: Introducción.	11
1.1 Motivación del proyecto.	12
1.2 Objetivos del proyecto.	12
1.3 Fases del proyecto.	13
1.4 Recursos utilizados.	14
1.5 Estructura y organización de la memoria.	14
Capítulo 2: Estado del arte.	16
2.1 Historia de la robótica.	17
2.2 Tipos de robots.	20
2.3 Robots móviles.	21
2.4 Robots teleoperados.....	23
Capítulo 3: Elementos del proyecto.	26
3.1 Raspberry Pi B+.	27
3.1.1 Puertos GPIO.	28
3.1.2 Software.	30
3.2 Sensores de luz.	31
3.3 Cámara.	32
3.4 Conexión Bluetooth.	33
3.5 Mando de teleoperación.....	34
3.6 Otros elementos del proyecto.....	37
3.6.1 Motores.	37
3.6.2 Ruedas.....	38
3.6.3 LED's.	39
Capítulo 4: Sistema propuesto.	41
4.1 Sistema mecánico.	42



4.1.1	Consideraciones para el diseño del modelo.....	42
4.1.2	Diseño de las piezas.	43
4.1.3	Fabricación y montaje.	47
4.2	Sistema electrónico.....	49
4.2.1	Placa 1.	51
4.2.2	Placa 2.	56
4.3	Programación.	64
4.3.1	Descarga del Sistema Operativo.	64
4.3.2	Instalaciones necesarias.....	64
4.3.3	Código.	68
4.3.4	Funciones del mando de teleoperación.	76
Capítulo 5: Pruebas y cambios.....		81
5.1	Prueba del sistema mecánico.....	82
5.2	Prueba del sistema electrónico.	82
5.3	Pruebas de programación.	84
Capítulo 6: Conclusión.		85
6.1	Conclusiones sobre el proyecto.....	86
6.2	Trabajos futuros.....	87
Capítulo 7: Presupuesto.		88
7.1	Presupuesto general.	89
7.2	Fechas de desarrollo.	90
Capítulo 8: Anexos.....		91
8.1	Código completo.....	92
8.2	Glosario.	95
8.3	Bibliografía.....	96



Índice de figuras

<i>Figura 1. Brazo mecánico creado por los griegos.</i>	17
<i>Figura 2. León mecánico de Leonardo de Vinci.</i>	17
<i>Figura 3. A la izquierda, Carel Kapel. A la derecha, Isaac Asimov.</i>	18
<i>Figura 4. Primer robot industrial.</i>	19
<i>Figura 5. Robot Curiosity en la superficie de Marte.</i>	21
<i>Figura 6. Robot con patas.</i>	22
<i>Figura 7. Robot con cadenas.</i>	22
<i>Figura 8. Chasis de un robot con ruedas.</i>	23
<i>Figura 9. Ejemplos de robots teleoperados.</i>	24
<i>Figura 10. Robot Curiosity.</i>	25
<i>Figura 11. Raspberry Pi B+.</i>	27
<i>Figura 12. Puertos y componentes de la placa Raspberry Pi B+.</i>	28
<i>Figura 13. Función de los puertos GPIO.</i>	29
<i>Figura 14. Logo de Raspbian.</i>	30
<i>Figura 15. Sensor de luz LDR.</i>	31
<i>Figura 16. Cámara Eye Toy.</i>	32
<i>Figura 17. Logo de Bluetooth.</i>	33
<i>Figura 18. Wii Remote.</i>	35
<i>Figura 19. Botones del mando Wii Remote.</i>	36
<i>Figura 20. Motor.</i>	38
<i>Figura 21. Ruedas.</i>	39
<i>Figura 22. LED blanco y LED verde.</i>	40
<i>Figura 23. Base inferior.</i>	43
<i>Figura 24. Balda superior o techo.</i>	44
<i>Figura 25. Pared lateral.</i>	44
<i>Figura 26. Balda interior.</i>	45
<i>Figura 27. Rueda de apoyo.</i>	46
<i>Figura 28. Chasis mecánico del robot.</i>	47
<i>Figura 29. Imagen final del robot.</i>	49
<i>Figura 30. Esquema general del sistema electrónico.</i>	50
<i>Figura 31. Circuito del motor 1.</i>	52
<i>Figura 32. Circuito del motor 2.</i>	53
<i>Figura 33. Circuito de la linterna.</i>	54
<i>Figura 34. Esquema de la placa 1.</i>	55
<i>Figura 35. Imagen final de la placa 1.</i>	56
<i>Figura 36. Circuito inicial del modo automático.</i>	57
<i>Figura 37. Modificación del circuito inicial del modo automático.</i>	59
<i>Figura 38. Circuito final del modo automático (para el motor 1).</i>	61
<i>Figura 39. Circuito final del modo automático (para el motor 2).</i>	62
<i>Figura 40. Esquema de la placa 2.</i>	63



<i>Figura 41. Imagen final de la placa 2.....</i>	<i>63</i>
<i>Figura 42. Diagrama de flujo de ambos motores.</i>	<i>78</i>
<i>Figura 43. Diagrama de flujo de la linterna.</i>	<i>79</i>
<i>Figura 44. Diagrama de flujo del modo automático.</i>	<i>80</i>



Índice de tablas

<i>Tabla 1. Tipos de robots según su arquitectura.</i>	<i>20</i>
<i>Tabla 2. Clases de Bluetooth.</i>	<i>33</i>
<i>Tabla 3. Piezas construidas.</i>	<i>47</i>
<i>Tabla 4. Lista de componentes mecánicos y electrónicos necesarios.</i>	<i>48</i>
<i>Tabla 5. Lista de componentes de la fuente de alimentación</i>	<i>48</i>
<i>Tabla 6. Lista de otros componentes.</i>	<i>48</i>
<i>Tabla 7. Código inicial.</i>	<i>70</i>
<i>Tabla 8. Cabecera.</i>	<i>71</i>
<i>Tabla 9. Comprobación de conexión mando-Raspberry Pi B+.</i>	<i>72</i>
<i>Tabla 10. Comprobación para desconexión.</i>	<i>72</i>
<i>Tabla 11. Código BOTÓN 2.</i>	<i>73</i>
<i>Tabla 12. Código BOTÓN UP.</i>	<i>73</i>
<i>Tabla 13. Código BOTÓN DOWN.</i>	<i>73</i>
<i>Tabla 14. Código ausencia de botón pulsado.</i>	<i>74</i>
<i>Tabla 15. Código BOTÓN 1.</i>	<i>74</i>
<i>Tabla 16. Código BOTÓN +.</i>	<i>74</i>
<i>Tabla 17. Código BOTÓN -.</i>	<i>75</i>
<i>Tabla 18. Código BOTÓN A.</i>	<i>75</i>
<i>Tabla 19. Código BOTÓN B.</i>	<i>76</i>
<i>Tabla 20. Funciones del mando de teleoperación.</i>	<i>77</i>
<i>Tabla 21. Presupuesto general.</i>	<i>89</i>
<i>Tabla 22. Código completo.</i>	<i>95</i>





Capítulo 1: Introducción.

En este capítulo se expondrá una pequeña introducción al proyecto realizado. En él se hablará de la motivación y de los objetivos del trabajo, así como de las fases en las que se dividió el proyecto y de los recursos utilizados.



1.1 Motivación del proyecto.

El presente proyecto engloba el diseño, la construcción y la implementación de un robot móvil autónomo cuya principal función es la de ser teleoperado por un usuario, el cual podrá observar por donde se mueve el robot por medio de un dispositivo con conexión a Internet, como por ejemplo, un Smartphone.

Esta teleoperación se realiza gracias a la conexión bluetooth de un mando de teleoperación con la placa controladora del robot (y su adaptador bluetooth). Gracias a este tipo de conexión, el usuario obtiene una gran libertad, ya que para manejar y mover el robot no tiene por qué estar conectado a través de un cable de conexión entre la placa y el mando. Así, por ejemplo, el usuario podrá mover el robot estando en una habitación distinta al robot.

El usuario controla todas las opciones disponibles del robot a través del citado mando. Así, puede manejar el robot manualmente o si lo prefiere, activar y desactivar el modo automático. Este modo hará que el robot detecte la luz a través de unos sensores y vaya hacia ese haz de luz detectado.

Los sensores utilizados para detectar la luz son unas resistencias LDR, las cuales, a través de los circuitos electrónicos diseñados, enviarán la información necesaria para activar a uno o a ambos motores y que estos hagan moverse al robot.

La principal motivación para la realización de dicho proyecto es la de diseñar, fabricar y construir un pequeño robot móvil de fácil utilización y abierto a trabajos futuros.

1.2 Objetivos del proyecto.

La finalidad de este proyecto es diseñar un robot y teleoperarlo desde un dispositivo electrónico con conectividad a internet. Se utilizará una Raspberry Pi como “cerebro” del robot y como mando de teleoperación, se utilizará un mando de la consola Wii de Nintendo.

Para poder ver que obstáculos se encuentran en su camino, al robot se le incorporará una cámara, gracias a la cual podremos ver a través de streaming en la pantalla de nuestro dispositivo electrónico.

También el robot llevará incorporada una linterna que se podrá activar y desactivar cuando el usuario lo decida. Pudiendo así el robot entrar en sitios oscuros.

Por último, el robot incorporará una opción para dejar a un lado la teleoperación y guiarse a través de la luz, es decir, que si sus sensores detectan luz, el robot irá hacia ella.



Por lo tanto, los objetivos de este proyecto los podríamos resumir en los siguientes 4 puntos:

- Conseguir una comunicación continua entre la cámara y la Raspberry Pi y que esta última sea capaz de emitir por streaming lo que la cámara ve.
- Conseguir una perfecta comunicación entre la Raspberry Pi y el mando de la consola Wii y hacer que las ordenes que emita éste último lleguen a realizarse.
- Configurar la electrónica para el movimiento de los motores de las ruedas y también para realizar la linterna.
- Realizar un circuito eléctrico para que el robot sea sensible a la luz y vaya hacia ella.

1.3 Fases del proyecto.

La realización de este proyecto se dividió en tres partes. A continuación se explica el trabajo realizado en cada una de estas fases:

- **Fase 1. Planificación:**
 - **Estudio de la tecnología a utilizar:** Estudio del lenguaje Python, utilizado en los ficheros ejecutables del proyecto.
 - **Estudio electrónico:** Estudio de toda la electrónica que será utilizada en el proyecto, como son los circuitos utilizados en la linterna o en los motores entre otros.
 - **Estudio mecánico:** Estudio de los parámetros físicos del robot como son la altura, ancho y largo.
- **Fase 2. Desarrollo y diseño:**
 - **Instalación:** Instalación del software *Raspbian* en la Raspberry Pi. Conexión mediante puerto USB de la cámara Eye Toy. Conexión mediante bluetooth con el mando Wii que teleopera con el robot.
 - **Electrónica:** Realización de los circuitos previamente estudiados.
 - **Desarrollo:** Pruebas de funcionamiento antes de montar el robot.
 - **Pruebas:** Pruebas realizadas antes de pasar el punto de no retorno en el montaje, para saber si el modelo mecánico funciona bien.
 - **Diseño:** Realización del diseño previamente estudiado después de ser confirmado en las pruebas.
 - **Simulación:** Estudio de los resultados del robot completamente montado.



- **Fase 3. Documentación:**
 - **Memoria del Trabajo de Fin de Grado:** Redacción del documento presente.
 - **Presentación:** Exposición audiovisual que será utilizada en la defensa de este proyecto ante el tribunal.

1.4 Recursos utilizados.

Para la realización de este proyecto, han sido necesarios los siguientes medios y materiales:

- Un dispositivo electrónico con conexión a internet y con el navegador Mozilla Firefox instalado (ya sea un ordenador, tablet, smartphone,...).
- Raspberry Pi B+.
- Cámara con conexión USB.
- Adaptador bluetooth USB.
- Dos motores.
- Dos ruedas.
- Dos fotorresistencias.
- 6 leds blancos.
- Un mando de la consola Wii.
- Una batería.
- Componentes electrónicos (resistencias, transistores y cables).
- Otros materiales (tornillos, contrachapado, soldador y estaño).

1.5 Estructura y organización de la memoria.

En este apartado se realizará una pequeña explicación de cada capítulo para una mejor lectura y comprensión del documento. La memoria consta de seis capítulos:

Capítulo 1: Introducción. Es el capítulo que se está leyendo en este instante. Trata sobre una pequeña introducción al trabajo, detallando objetivos, fases y motivación del proyecto, así como los recursos utilizados.

Capítulo 2: Estado del arte. Se tratarán varios temas relacionados con el presente de la robótica, Comenzará con la historia de la robótica, continuando por los tipos de robots existentes. A continuación se hará más hincapié en los robots móviles y teleoperados.

Capítulo 3: Elementos del proyecto. En este apartado se describirán aquellos elementos que son necesarios para la realización del presente proyecto.



Capítulo 4: Sistema propuesto. En este capítulo se explicará la mecánica, electrónica y programación que se ha llevado a cabo en el robot. En la parte mecánica se explicará el modelo elegido, en la parte electrónica se verán todas las conexiones que han sido necesarias para el correcto funcionamiento del robot y en la parte de programación se verá el código realizado, así como la explicación de la utilización del mando y los correspondientes diagramas de flujo.

Capítulo 5: Pruebas. Se desarrollan las distintas pruebas llevadas a cabo.

Capítulo 6: Conclusión. Se explicarán y analizarán las conclusiones sobre el trabajo y se hablará de trabajos futuros.

Capítulo 7: Presupuesto. Se realizará un presupuesto de lo que ha costado la realización del proyecto.

Adicionalmente en el capítulo 8, se anexan un conjunto de apéndices donde se muestran algunos aspectos como el código completo que está programado en el robot, el glosario y la bibliografía.



Capítulo 2: Estado del arte.

En este capítulo se verá una pequeña introducción a la historia de la robótica. Después se hablará de los distintos tipos de robots que hay para acabar explicando qué son los robots móviles y los robots teleoperados.

2.1 Historia de la robótica.

La palabra robot tiene varios significados dependiendo siempre de la persona a la que se le pregunte. Muchas veces, el significado que se le da a esta palabra dista mucho de la realidad y de lo que la ciencia a día de hoy puede llegar a conseguir. En parte, esto se debe a la gran influencia que han tenido sobre las personas los libros, películas y series de ciencia ficción.

Desde hace muchos siglos, el hombre se ha interesado por desarrollar artificialmente partes de nuestro cuerpo. En la época de los imperios egipcio y griego se empezaron a dar los primeros pasos, construyendo artilugios capaces de emular las funciones y los movimientos de los seres vivos. Estos mecanismos (los griegos los llamaban *automato*, origen de la palabra *autómata*) se utilizaban para impresionar a los adoradores de los templos y para demostrar a los dioses su devoción.

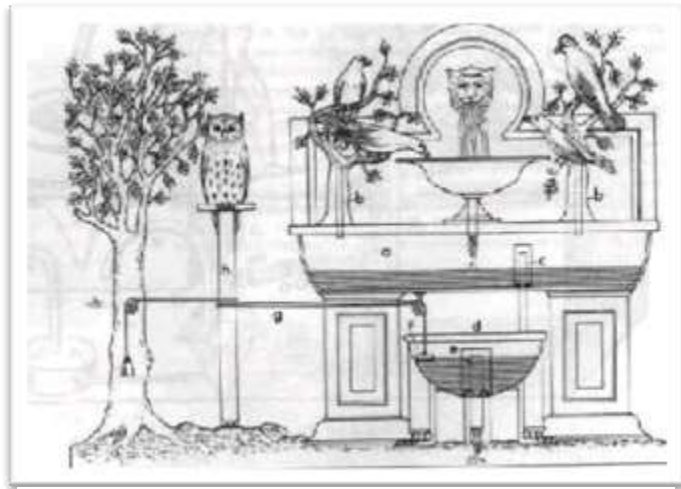


Figura 1. Brazo mecánico creado por los griegos.

Los años y los siglos fueron pasando y el hombre seguía desarrollando la idea de crear vida artificial. Incluso el grandísimo Leonardo da Vinci hizo sus propias creaciones en la época del Renacimiento como el “León mecánico” [2]. Trataba de un mecanismo de tamaño real capaz de caminar sin estar provisto ni de motores ni de electricidad, simplemente con engranajes mecánicos.



Figura 2. León mecánico de Leonardo da Vinci.

Entrando ya a mediados del siglo XVIII, gracias a la Revolución Industrial, se fueron realizando nuevas creaciones que pudieran aportar más rapidez a la fabricación. Algunos transformadores de energía, como la máquina de vapor o el molino de agua, fueron sustituyendo a la fuerza humana. Así la fabricación era más fluida y la calidad

de vida aumentaba, ya que los muchos hombres pasaron a trabajar en las fábricas logrando así tener un trabajo menos duro.

Llegaba el siglo XX y con él el gran avance tecnológico del siglo: la robótica. Fue en el año 1921 cuando el escritor checo Karel Capek [1] utilizó por primera vez la palabra *robot*. Lo hace en su obra R.U.R. (Robots Universales Rossum), un drama que trata sobre una empresa que crea humanos artificiales orgánicos capaces de aligerar la carga de trabajo de otras personas. El término lo creó el hermano del escritor checo, Josef Capek, partiendo de la palabra *robota*, que en checo



significa “trabajo”, sustituyendo a la palabra *automat*, que utilizó en obras anteriores. Fácilmente el término podría haber quedado en el olvido, pero gracias a otros escritores de ciencia ficción, siendo el más destacado Isaac Asimov, asiduos a escribir obras sobre la interacción de los humanos con los robots, el término siguió usándose y siendo cada vez más conocido. En estas obras, los robots estaban dotados de cerebros positrónicos, en los que estaban codificadas las tres leyes de la robótica [1]. Estas leyes fueron creadas y redactadas por Isaac Asimov y que surgen como medida de protección de los seres humanos ante los robots:

Figura 3. A la izquierda, Carel Kapel. A la derecha, Isaac Asimov.

1. Un robot no hará daño a un ser humano o, por inacción, permitir que un ser humano sufra daño.
2. Un robot debe obedecer las órdenes dadas por los seres humanos, excepto si estas órdenes entrasen en conflicto con la 1ª Ley.
3. Un robot debe proteger su propia existencia en la medida en que esta protección no entre en conflicto con la 1ª o la 2ª Ley.

A mediados de este siglo surgió la cibernética [3]. El término se refiere a la ciencia que estudia los sistemas de comunicación y de regulación automática de los seres vivos y los aplica a sistemas electrónicos y mecánicos que se parecen a ellos. Así, las industrias también evolucionaron y se empezaron a basar en la ciencia. Gracias a ello los avances tecnológicos en el campo de la electrónica llegaron y se creó el ordenador. Esto supuso un punto de inflexión, ya que esta nueva creación era capaz de procesar y comunicar los datos con más rapidez. Aquí pasó lo mismo que en la Revolución Industrial, y es que cambió la forma de vivir, pensar e incluso de actuar de la sociedad. Desde

entonces se necesitan menos trabajadores de producción y se precisan más trabajadores informáticos.

Fue en 1961 cuando fue instalado el primer robot industrial. La creación la llevó a cabo la empresa Unimation (Universal Animation) fundada por Geroge Devol y Joseph F. Engelberger en 1956. Para este primer brazo robótico se basaron en modelos y patentes originales de Devol. En un principio estos brazos fueron apodados como *máquinas de transferencia programables*, ya que su función era única y exclusivamente la de transportar objetos de un punto a otro a menos de tres metros de distancia. Pero con el avance de la ciencia se fueron creando nuevos y mejores brazos, hasta llegar a los actuales, los cuales son capaces de hacer las tareas de pintar, cargar y descargar, soldar, montaje electrónico, etc., que proporcionan a las industrias una mayor producción y una mejora de calidad en los productos [4].



Figura 4. Primer robot industrial.

Pero aun así, a día de hoy, encontramos en los diccionarios acepciones incompletas para el término *robot*. El ejemplo más claro es el de la Real Academia Española (R.A.E.), que define este término como “máquina o ingenio electrónico programable, capaz de manipular objetos y realizar operaciones antes reservadas solo a las personas” [5]. Como esta acepción es insuficiente y no logra abarcar toda aquella idea que las personas asociamos a la palabra robot, utilizamos un segundo término para definir a cada uno de ellos. Este segundo vocablo puede ser el medio en el que opera o, simplemente, algunas de sus aplicaciones o su forma. Algunos de los ejemplos son los siguientes: robot aéreo, robot submarino, robot móvil, robot humanoide o robot de entretenimiento [1].



2.2 Tipos de robots.

Existen una gran cantidad de robots y cada uno de ellos tiene sus propias características como arquitectura, movilidad, aplicaciones, etc., por lo que no es posible hacer una sola clasificación que incluya todas las propiedades. Una de las formas más populares de clasificar a los robots es por su arquitectura:

TIPOS DE ROBOTS SEGÚN SU ARQUITECTURA	
Poliarticulados	Robots industriales o manipuladores
Móviles	Terrestres, acuáticos, aéreos, espaciales
Androides	Robots humanoides de complejo diseño
Zoomórficos	Robots que representan los sistemas de locomoción de diversos seres vivos
Híbridos	Robot formado por combinación de dos o más de las arquitecturas anteriores

Tabla 1. Tipos de robots según su arquitectura.

Clasificando a los robots mediante otro criterio, como por ejemplo su aplicación, destacan los robots especializados en el campo de la medicina. Existen los robots quirúrgicos, capaces de aumentar la precisión de un cirujano, ya que elimina los temblores humanos, es capaz de entrar a zonas inasequibles para los médicos y daña menos tejido sano en los alrededores de la región afectada. También existen otros tipos de robots como los robots de limpieza, que ahorran a una persona el trabajo de barrer y limpiar, los robots de rehabilitación médica, que como su propio nombre indica, ayuda a un paciente a su rehabilitación después de una intervención o robots de asistencia personal, los cuales ayudan a pacientes ancianos o personas discapacitadas a realizar acciones que por sí mismas no pueden realizar.

Otros tipos de robots son los robots de campo, que suelen trabajar en exteriores. Para estos tipos de robots se presentan grandes retos. El primero de ellos es que tengan un correcto funcionamiento de forma segura ante la presencia de personas, ya que en su mayoría estos tipos de robots tienen una gran interacción con los seres humanos. Otro de los objetivos es el de moverse y operar en un entorno desordenado y complejo. Un claro ejemplo son los robots enviados a Marte [6], que antes de ser enviados al planeta rojo deben de aprobar unas pruebas bastante exigentes, ya que la N.A.S.A, empresa que transporta el robot a Marte, no se puede arriesgar a gastarse una importante

cantidad de dinero para enviar un robot que al poco tiempo tenga un accidente o, simplemente, no sea capaz de moverse por la superficie de dicho planeta.



Figura 5. Robot Curiosity en la superficie de Marte.

2.3 Robots móviles.

El presente proyecto trata sobre un pequeño robot móvil, por lo que a continuación se entrará un poco más en detalle sobre este tipo de robots. Los robots móviles no se ciñen a una posición fija y deben de ser capaces de moverse por cierto tipo de entorno. Así, un robot puede estar diseñado para ser acuático (flotante o submarino), aéreo o terrestre.

Hablando del tipo de locomoción terrestre, que será la que se proporcionará al robot del presente proyecto, existen varios tipos entre los que destacan tres de ellos: patas, cadenas y ruedas.

Los robots con patas [7] superan con mayor facilidad aquellas dificultades que el terreno pueda ocasionar al robot (un claro ejemplo son los robots capaces de subir escaleras). Este tipo de locomoción puede dar una omnidireccionalidad que otros tipos no dan, a la vez que permiten que el rozamiento sea menor. En su contra juega

que es bastante complicado encontrar una buena estabilidad para el robot, que su construcción es bastante más compleja, ya que cada pata necesita como mínimo un par de motores, produciéndose así un mayor coste y una mayor complejidad en los algoritmos de control y que consumen más energía. Por todo ello los sistemas de locomoción por patas son un campo de investigación muy activo.



Figura 6. Robot con patas.

Los vehículos con cadenas se suelen utilizar para robots que trabajan en un entorno con terrenos irregulares. Permiten superar obstáculos mayores y sufren menos daños por el entorno que las ruedas. También su manejo es sencillo. En contrapartida aparece a la hora de realizar un giro, ya que consume mucha potencia y no dispone un modelo preciso para ese tipo de movimiento.



Figura 7. Robot con cadenas.

Los vehículos con ruedas [8] son los más populares y los más usados por varias razones. Una de ellas es que son más sencillos de construir. También que son económicamente más rentables y que pueden transportar una mayor carga que los vehículos con cadenas o patas. Su mayor inconveniente surge a la hora de superar obstáculos, ya que estos tipos de robots no podrán superar obstáculos que superen en

altura el radio de sus ruedas, lo que hace que el comportamiento en entornos con terreno irregular sea malo. Una solución es la de utilizar ruedas con un mayor diámetro, pero a veces, esta solución no es práctica.



Figura 8. Chasis de un robot con ruedas.

De estos tipos de locomoción descritos para robots, se ha optado por escoger un vehículo con ruedas, ya que cumple con los requisitos funcionales y económicos requeridos para el presente proyecto.

2.4 Robots teleoperados.

La telerrobótica es el área de la robótica que estudia el control de robots desde la distancia, principalmente usando conexiones wireless (como Wi-Fi, Bluetooth, la Red del Espacio Profundo, y similares), conexiones "ancladas", o a través de Internet. Es una combinación de dos campos importantes, teleoperación y telepresencia.

El significado de la palabra *telepresencia* es el de “sentir como si estuvieras en algún otro lugar”. Trasladando el término al presente proyecto, quiere decir que el usuario debe de sentirse como si estuviera en la posición del robot. Por otro lado, el término *teleoperación* significa “hacer el trabajo a distancia”. Es una definición compleja, ya que “trabajo” y “distancia” pueden referirse a casi cualquier cosa. Un ejemplo es el robot *Roomba* [9], que su “trabajo” es el de aspirar el suelo de la casa. En el caso de la palabra “distancia”, se puede referir a una gran distancia física (el robot *Curiosity* en Marte [6]) o a un simple cambio de escala (cirujía robótica).

Los robots teleoperados se utilizan para llevar a cabo operaciones que el ser humano no puede realizar, ya sea por su complejidad o debido a que la operación se ha de realizar en un entorno demasiado hostil.



Figura 9. Ejemplos de robots teleoperados.

Un sistema teleoperado es todo aquel sistema que nos permite gobernar un robot situado en una zona lejana respecto a su mando de teleoperación, controlando así su movimiento. El individuo que tiene la posesión del mando de teleoperación pulsará los botones correspondientes, enviando así la orden al robot. De este modo, el movimiento del robot, siempre dependerá de lo que quiera la persona que lo maneja. El sistema de operación debe de ser de fácil entendimiento para que el robot pueda ser dirigido por el mayor número de personas posibles, así pues, si el sistema de operación es muy complejo, sólo un número muy reducido de personas lo podrían utilizar y eso no es conveniente.

Un sistema de teleoperación consta de los siguientes elementos [10]:

- **Operador o teleoperador:** persona que realiza las operaciones del robot a distancia. Su intervención en la manipulación del robot puede ser puntual o de control continuo.
- **Dispositivo teleoperado:** Dispositivo controlado por el operador y que trabaja en una zona remota.
- **Interfaz:** Conjunto de dispositivos que permiten que haya una comunicación entre el operador y el sistema de teleoperación.



- **Control y canales comunicación:** Conjunto de dispositivos que permiten la comunicación entre el teleoperador y el dispositivo teleoperado. Para que esto pueda ocurrir, este conjunto de dispositivos modulan, transmiten y adaptan las señales que se envían.
- **Sensores:** Conjunto de dispositivos que nos da información del entorno, creando una señal eléctrica para que sea utilizada por el interfaz y el control.

Este campo de los robots teleoperados ha estado muy presente en lo que a desarrollo tecnológico se refiere. Y es que el ser humano ha pasado de teleoperar un simple brazo robótico a mandar robots a otro planeta, como es el caso del *Curiosity* [6], enviado por la N.A.S.A. a Marte, para saber qué hay allí, siendo teleoperado desde nuestro planeta.



Figura 10. Robot Curiosity.



Capítulo 3: Elementos del proyecto.

En este apartado de la memoria se explicarán los distintos componentes que se han utilizado para llevar a cabo el presente proyecto.

El más importante, la placa controladora, la Raspberry Pi B+, que hará de “cerebro” del robot. También se hablara del software, del tipo de conexión utilizada y de los demás componentes empleados.

3.1 Raspberry Pi B+.

Raspberry Pi B+ [11] es un ordenador de placa reducida o placa única de bajo costo, desarrollado en Reino Unido por la Fundación Raspberry Pi. Algunas de las características de esta placa son:

- Procesador: Broadcom BCM2835 SoC full HD
- RAM: 512 MB SDRAM 400 MHz
- Almacenamiento: tarjeta microSD
- USB: cuatro puertos USB 2.0
- Energía: 600mA hasta 1,8 A a 5 V
- Pines GPIO: 40



Figura 11. Raspberry Pi B+.

La Raspberry Pi B+ se ha utilizado como el cerebro del robot, es decir, el dispositivo que se encarga de la toma de decisiones, de qué pasa si está activado el modo automático (el robot va hacia la luz) o si está en modo teleoperativo si pulsamos uno u otro botón del mando.

Se ha utilizado la Raspberry Pi ya que tiene un precio económico y es fácil de utilizar. Otra de las ventajas que tiene es que al estar basado en código abierto de Linux, el usuario puede tener acceso a software y descargas gratuitas que le ayudarán. También cuenta con una gran comunidad de usuarios de este dispositivo y es un dispositivo muy flexible para el usuario, ya que puede tener muchas funciones.



Se eligió el modelo Raspberry Pi B+, porque, aparte de que conservaba las utilidades de los modelos anteriores, posee una ventaja respecto a estos, y es que tiene una cantidad mayor de pines digitales. Estos serán utilizados pero no en su totalidad, así se creará la posibilidad de una futura expansión añadiendo más sensores.

En la siguiente imagen se muestran las distintas partes y puertos que componen la placa Raspberry Pi B+:

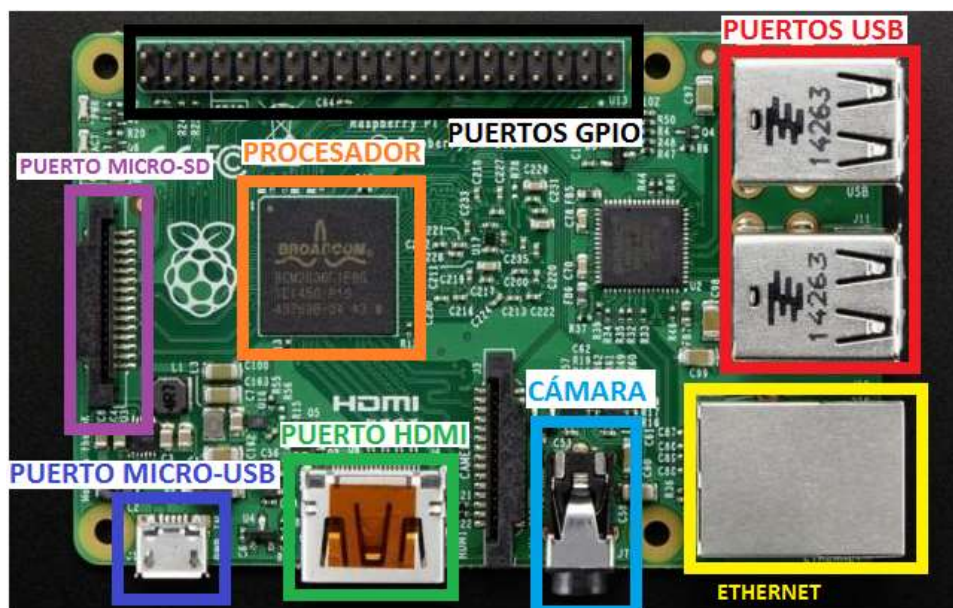


Figura 12. Puertos y componentes de la placa Raspberry Pi B+.

3.1.1 Puertos GPIO.

Las siglas GPIO significan General Purpose Input/Output, lo que traducido al castellano significa “propósito general de entrada/salida”. Son pines genéricos que pertenecen a un chip y pueden ser programados por una persona, ya sean utilizados como entrada o como salida.

Los pines GPIO:

- Pueden ser configurados para ser entrada o para funcionar como salida.
- Pueden ser activados o desactivados.



- Se pueden leer sus valores de entrada (alto es 1 y bajo es 0).
- Se pueden leer o escribir sus valores de salida.
- Valores de entrada que a menudo se utilizan como IRQ. (normalmente en eventos de activación).
- El dispositivo puede utilizar DMA para mover cantidades de datos grandes de una manera eficiente, tanto dentro como fuera del dispositivo.

El modelo de Raspberry Pi escogido trae 40 pines de este tipo. La función de cada pin se puede comprobar en la siguiente imagen:

Key			
Power +		UART	
GND		SPI	
PC		GPIO	

3.3V	1	2	5V
GPIO2	3	4	5V
GPIO3	5	6	GND
GPIO4	7	8	GPIO14
GND	9	10	GPIO15
GPIO17	11	12	GPIO18
GPIO27	13	14	GND
GPIO22	15	16	GPIO23
3.3V	17	18	GPIO24
GPIO10	19	20	GND
GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8
GND	25	26	GPIO7
DNC	27	28	DNC
GPIO5	29	30	GND
GPIO6	31	32	GPIO12
GPIO13	33	34	GND
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
GND	39	40	GPIO21

Figura 13. Función de los puertos GPIO.

Estos puertos representan la interfaz entre la Raspberry Pi y su entorno. Los pines no disponen de buffers que los protejan, por lo tanto, el usuario ha de tener cuidado con los voltajes y las intensidades utilizadas para no dañar la placa.

Como se observa en la anterior imagen, no todos los pines tienen la misma función. Se pueden distinguir cuatro grupos distintos de pines:

- **Pines de alimentación:** El dispositivo Raspberry Pi B+ tiene 12 pines de alimentación:



- Dos de ellos (el pin 1 y el 17) son capaces de proporcionar 3,3 V, limitados a 50 mA.
- Los pines 2 y 4 proporcionan una salida de 5 V.
- Los ocho pines restantes (6, 9, 14, 20, 25, 30, 34 y 39) son pines de tierra.
- **DNC:** Sus siglas tienen como significado “Do Not Connect” es decir, “no conectar”. En este momento son pines sin funciones.
- **GPIO normales:** Pines programables.
- **GPIO especiales:** Pines que sirven para conexiones especiales como a una interfaz UART u otros tipos como I2C, SPI, SDA, MOSI, MISO, etc

3.1.2 Software.

El software utilizado e instalado en la Raspberry Pi B+ ha sido Raspbian [12], un sistema operativo libre y gratuito basado en Linux en su distribución Debian. Este sistema operativo está optimizado para ser instalado y usado en Raspberry Pi. Como definición de sistema operativo, se podría decir que es un conjunto de programas básicos y utilitarios que permiten que, en este caso, la Raspberry Pi, haga algo útil. Pero Raspbian tiene algo más que esto. Y es que contiene aproximadamente unos 35.000 paquetes precompilados, de tal forma que la instalación del paquete deseado sea sencilla.



Figura 14. Logo de Raspbian.

La imagen de este sistema operativo se puede descargar directamente y sin ningún coste para el usuario que se la descargue en la página de Raspberry Pi en la sección de descargas.



El software utiliza LXDE como escritorio y, como navegador web, Midori. Además, tiene IDLE, que es un entorno de desarrollo integrado para Python. Permite configurar el sistema operativo sin la necesidad de modificar archivos de configuración manualmente, sino que con el menú “raspi-config” se puede lograr.

3.2 Sensores de luz.

Hay un amplio abanico de sensores lumínicos. El elegido en este proyecto es una fotorresistencia LDR (Light-Dependent resistor).

Una fotorresistencia podría ser definida como una resistencia cualquiera con la característica de cambiar su valor dependiendo de la cantidad de luz que lo ilumina [13]. De esta forma, cuanta más luz incida sobre ella, menor será la resistencia que oponga a la electricidad. De forma análoga, cuanta menos luz incida sobre el sensor, mayor será la resistencia que oponga el dispositivo, dejando así que pase menos electricidad por ella.



Figura 15. Sensor de luz LDR.

Los materiales utilizados para su fabricación son el sulfuro de talio, el sulfuro de cadmio, el sulfuro de plomo, y el seleniuro de cadmio. Las fotorresistencias constan de un cuerpo formado por una célula y dos patillas. El cuerpo está formado por sulfuro de cadmio. Este sulfuro es un semiconductor y es gracias a él por el cual la resistencia de este elemento cambia de valor dependiendo de la cantidad de luz que incide sobre ella.

La función que desarrollará este componente electrónico en el presente proyecto será la de captar la cantidad de luz que llega al robot y, en función de ello, encender los LED's y los motores correspondientes. Es decir, que si el robot entra en una habitación oscura y al sensor colocado en la parte derecha del robot le llega un haz de luz, pero al sensor situado en la parte izquierda no le llega nada, el circuito electrónico se



encargará de encender el motor correspondiente para que el robot gire hacia ese lado y vaya hacia la luz.

3.3 Cámara.

La cámara utilizada para este proyecto, es una cámara de Sony llamada Eye Toy. Ella hará la misma función que los ojos hacen en nuestro cuerpo. Las cámaras captan una gran información sensorial y, gracias a ella, se podrá saber qué rodea a nuestro robot. Para dotar de una información útil a nuestro robot, se debe de procesar la imagen y aquí es donde nos encontramos el problema principal de este tipo de sensores, y es su velocidad de procesamiento.



Figura 16. Cámara Eye Toy.

Esta cámara es sensible al movimiento. Cuando detecta un objeto que se mueve, emite la imagen con el objeto móvil detectado metido en un rectángulo.

Su conexión con la Raspberry Pi se realiza por medio de uno de los puertos USB que esta contiene.

La cámara tiene un pequeño soporte en su inferior que permite que la rotación de su eje vertical. También este mismo soporte permite que la cámara se pueda mover hacia arriba o hacia abajo pero, en el caso de este proyecto, estará fija, ya que lo que queremos es que la cámara nos emita imágenes de lo que el robot tiene delante de él.

Como conclusión, una cámara proporciona mucha información en comparación con otros sensores. Como desventaja está la velocidad de procesamiento, ya que al ser



tanta información, a veces no es tan útil y se necesita mucho tiempo para los cálculos de los algoritmos que se utilicen. Como gran ventaja, nos proporciona que no tendremos por qué fiarnos de unas señales captados por algún tipo de sensor, sino que podremos ver con nuestros propios ojos qué rodea al robot, ya que la emisión de la cámara la podremos seguir en cualquier aparato electrónico con conexión a Internet.

3.4 Conexión Bluetooth.

El bluetooth es un sistema de ondas de radio de corto alcance, cuyo objetivo es simplificar las comunicaciones entre dispositivos informáticos. La función que tendrá en el presente proyecto será la de comunicar al mando de



Figura 17. Logo de Bluetooth.

teleoperación con la Raspberry Pi B+, es decir, con el robot. Para ello será obligatoria la utilización de un adaptador Bluetooth, que será insertado en uno de los puertos USB de la Raspberry Pi B+.

La comunicación por Bluetooth, entre otras muchas cosas, sustituye la tradicional comunicación por cable en equipos GPS y equipamiento médico, la transferencia de fichas de contactos y citas entre dispositivos electrónicos o, lo que interesa para la realización de este proyecto, estar presente en controles remotos y contar con mandos inalámbricos en consolas de juegos.

Las comunicaciones inalámbricas bluetooth y WI-FI son tecnologías complementarias, pero Bluetooth tiene la ventaja de simplificar el descubrimiento y configuración de los dispositivos, ya que estos pueden contactarse sin un control explícito de dirección de red, permisos y otros aspectos típicos de redes tradicionales.

Los dispositivos son clasificados por clases [14]:

Clase	Alcance (aproximado)	Potencia máxima permitida
Clase 1	≈ 100 metros	100 mW
Clase 2	≈ 10 metros	2,5 mW
Clase 3	≈ 1 metro	1 mW

Tabla 2. Clases de Bluetooth.

Para el presente proyecto se ha utilizado un adaptador de Bluetooth 4.0 de la marca Trust. Su unión con la Raspberry Pi B+ se simplifica a una conexión USB.



Se ha elegido este modelo por varias razones. Una de ellas es la cuestión económica, y es que este adaptador no tiene un precio muy elevado. También el hecho de que necesita un bajo uso energético para su correcto funcionamiento y que tiene una cobertura de 15 metros, lo que nos sirve para poder interactuar con el robot de una habitación a otra.

Bluetooth utiliza la técnica FHSS (Frequency Hopping Spread Spectrum, traducido al español como Espectro ensanchado por saltos de frecuencia). Esto consiste en dividir la banda de frecuencia de 2,402 – 2,480 GHz en 79 canales de 1 MHz de ancho cada uno (la denominación que se le puso a cada una de estas divisiones fue salto) y después, transmitir la señal utilizando una secuencia de canales que sea conocida tanto para la estación emisora como para la receptora. Por lo tanto, al cambiar de canales con una frecuencia de 1600 veces por segundo, el estándar Bluetooth puede evitar la interferencia con otras señales de radio.

El futuro de este tipo de comunicación está dirigido al aumento de su ancho de banda, permitiendo así alcanzar la sincronización y transferencia de grandes cantidades de datos entre dispositivos portables, impresoras, proyectores o televisores.

Por otro lado, el alcance de los distintos tipos de clases de Bluetooth, viene dado y probado en un entorno casi ideal, sin obstáculos que interfieran en la señal emitida y en perfectas condiciones. Sin embargo, el rango de alcance de un dispositivo Bluetooth, puede ser disminuido si nos encontramos con obstáculos como pueden ser muebles, paredes, la atmósfera o condiciones geográficas.

Principalmente, el sistema Bluetooth, está destinado a la comunicación de corto alcance. Por lo que, al igual que la mayoría de conexiones inalámbricas, cuanto más lejos estén ambos dispositivos, más débil y lenta será la señal. La ruta de comunicación entre dispositivos hay que tenerla lo más despejada posible, para que la señal viaje en las mejores condiciones posibles.

3.5 Mando de teleoperación.

El dispositivo elegido para hacer las funciones de mando de teleoperación, es un mando de la consola Wii, llamado *Wii Remote*.

Este mando contiene 11 botones. Cada botón tiene una función. Dependiendo del botón que sea pulsado, la Raspberry Pi recibirá una orden que será procesada por el programa creado.



Las principales características son la capacidad de detección de movimiento y la habilidad de apuntar hacia objetos en la pantalla.

Este mando es similar a un mando de control remoto de la televisión. En su cara anterior podemos diferenciar diez de sus once botones: la cruz de dirección (con cuatro botones: UP, DOWN, RIGHT, LEFT), botones A, +, HOME, -, 1 y 2. En su cara posterior, tan solo queda un botón llamado B, en forma de gancho.

El Wii Remote puede detectar la aceleración a lo largo de tres ejes. También incorpora un sensor óptico para saber dónde está apuntando. Otras de sus cualidades que destacan es que dispone de un sistema de audio, ya que el mando contiene un altavoz propio en su parte frontal. Como última característica a destacar, es la vibración, utilizada en el presente proyecto.



Figura 18. Wii Remote.



Figura 19. Botones del mando Wii Remote.



3.6 Otros elementos del proyecto.

Los robots deben de disponer de componentes electrónicos capaces de realizar las operaciones ordenadas por la unidad de control.

Un actuador es un dispositivo capaz de transformar energía hidráulica, neumática o eléctrica en la activación de un proceso con la finalidad de generar un efecto sobre un proceso automatizado. Este recibe la orden de un regulador o controlador y en función a ella genera la orden para activar un elemento final de control.

En este proyecto, se han utilizado dos tipos de actuadores: motores y LED's.

3.6.1 Motores.

Para este tipo de proyectos existen diversos tipos de motores. Para la realización de este robot, se han utilizado dos motores de corriente continua (DC). Normalmente los motores de corriente continua suelen tener mucha velocidad pero poca fuerza, por lo que nuestra elección debe ser la de un motor que sin una velocidad tan alta le pueda imprimir más fuerza al mecanismo.

Los motores elegidos para la realización del robot, son unos pequeños motores de metal con una gran potencia reductora. Las características [15] de cada uno de ellos son las siguientes:

- Sus dimensiones son:
 - largo: 24 mm
 - ancho: 10 mm
 - altura: 12 mm.
- Presenta un eje con un diámetro de 3,9 mm. Con ranura de bloqueo.
- El ratio de la reductora es 298:1.
- El voltaje nominal es 6 Vcc (puede funcionar con tensiones de 3 V a 9 V).
- Realiza una velocidad de giro de 100 revoluciones por minuto sin carga.
- Consume 70 mA sin carga. El máximo a consumir es de 1600 mA.
- Su par motor es 5 kg·cm.
- Su peso es de 10 gramos.

El funcionamiento del motor es sencillo, y es que cuando la señal eléctrica le llega, provoca que gire la primera rueda del engranaje, lo que deriva en los giros de los siguientes engranes. Así provoca que la última rueda del engranaje gire. La salida la tiene en el eje, al que se le une la rueda del robot que gracias al mecanismo ya explicado, puede girar y hace que el robot se pueda mover. Este pequeño motor eléctrico elegido para ambas ruedas del robot, cuenta con una gran potencia reductora.



Figura 20. Motor.

Este sistema de reducción permite reducir la velocidad de giro del motor para poder controlarla mejor, aumentando su par en la rueda de salida (última rueda del motor).

Ambos motores han de estar conectados con los puertos GPIO de la Raspberry Pi. El motor derecho lo hará al puerto GPIO4 o lo que es lo mismo, al pin 7. Por su parte, el motor de la rueda izquierda se conectará en el puerto GPIO17, equivalente al pin 11.

Si el puerto correspondiente a un motor está encendido (estado ON), el motor se activará gracias al circuito implementado, ya que le llegará la cantidad de corriente suficiente para que esto pase. Si por el contrario el estado del puerto GPIO correspondiente a uno de los motores es OFF o desactivado, el motor no se moverá ya que no le llega corriente a través del puerto de la placa.

3.6.2 Ruedas.

Las ruedas son un elemento indispensable para la realización de este proyecto.

Son necesarias una pareja de ruedas, una a cada lado del robot. Las elegidas tienen un diámetro de 86,5 mm y un grosor de 10 mm. Son compatibles con el par de motores utilizados, ya que el tamaño del eje de salida del motor coincide con el agujero del eje de las ruedas.



Las ruedas escogidas, con un diámetro de casi 10 cm, se podrían calificar como de gran tamaño. Con ellas se ganará velocidad, ya que con cada giro que realice se el robot recorrerá (en línea recta) algo más de 31 cm:

$$L = 2 \cdot \pi \cdot r = 2 \cdot \pi 5(\text{cm}) \cong 31,4 \text{ cm}$$

Con este tipo de ruedas, al ser tan altas, colocan el centro de gravedad del robot a una altura mayor que si el diámetro de las mismas no fuera tan grande, por lo que el robot perderá algo de estabilidad. Esto no tendrá una gran importancia en el caso del presente proyecto, ya que el robot podrá girar sin tener el usuario la preocupación de volcarlo.



Figura 21. Ruedas.

3.6.3 LED's.

Los LED's son componentes optoelectrónicos pasivos y, como sus propias siglas indican, un diodo emisor de luz (en inglés, Light-emitting diode). Su principal función es la de transformar la energía eléctrica en una fuente luminosa.

Su funcionamiento se basa en la emisión de fotones, o lo que es lo mismo la emisión de luz, cuando pasan los electrones que portan la electricidad a través del diodo. Este fenómeno es llamado electroluminiscencia. Los LED's, al ser diodos, solamente dejan pasar la electricidad en un sentido y son catalogados como componentes digitales de salida; digitales porque estos sólo tienen dos estados, el apagado (0) y el encendido (1), y de salida porque debemos de comandarlos nosotros y decidir en qué estado se

encuentra en cada momento, al contrario que los sensores, que son componentes que nos proporcionan la lectura de un valor.

Los LED's se usan en tecnologías avanzadas y de control, debido a su capacidad de operar a altas frecuencias, pero su uso más común, es el de indicador en dispositivos electrónicos. Así se han podido sustituir las pequeñas bombillas que se utilizaban por este componente electrónico.

Actualmente se están desarrollando las lámparas led para poderlas utilizar en el ámbito de la iluminación ambiental, ya que estas contienen algunas ventajas sobre las lámparas incandescentes y fluorescentes. Algunas de las ventajas son:

- Bajo consumo de energía.
- Vida útil más larga.
- Menor tamaño.
- Mayor robustez.
- Mayor resistencia a las vibraciones y a los golpes.
- Emisión de calor reducida.
- Encendido instantáneo.
- Reproducción de los colores con gran fidelidad.



Figura 22. LED blanco y LED verde.

Para el presente proyecto se han utilizado un total de 8 LED's como el de la imagen, 2 unidades de color verde y 6 de color blanco. Para su correcto funcionamiento, no hay más que conectarlos, alimentarlos, conectarlos a tierra y a su respectivo puerto GPIO de la Raspberry Pi B+. La función que realizarán los LED's blancos será la de una linterna, una pequeña lámpara hecha a base de los 6 LED's que nos permitirá iluminar la parte delantera del robot y así se podrá observar con mayor claridad lo que la cámara enfoca. Por último, la función que realizarán los dos LED's verdes estará presente sólo en el modo automático, y es que serán los encargados de avisarnos encendiéndose cuando su respectivo sensor lumínico capta un haz de luz.



Capítulo 4: Sistema propuesto.

En este capítulo se explicarán los distintos sistemas que tiene el robot, como son el mecánico, el electrónico y la programación.

En el sistema mecánico se describirán todos los pasos seguidos para la fabricación del chasis del robot, desde el material hasta la forma que tomará el robot.

En el sistema electrónico se verán todos los circuitos que se han realizado en el proyecto.

En la programación se mostrará todo el código, sentencias y descargas que han sido necesarias.



4.1 Sistema mecánico.

Para que el robot pueda cumplir los objetivos marcados, la elección del sistema mecánico es un elemento muy importante. En los siguientes apartados se detallarán los pasos seguidos hasta el montaje final, pasando por el diseño, materiales y montaje realizado.

4.1.1 Consideraciones para el diseño del modelo.

A continuación se explicarán las consideraciones que se han tenido en cuenta para tomar cada decisión en la fabricación del presente proyecto.

➤ Material utilizado.

El material utilizado ha sido madera de contrachapado. Su elección se fundamenta básicamente en que es un material duro, pero a la vez es manejable, tiene una baja densidad, es accesible económicamente y es fácil de encontrar en cualquier establecimiento especializado.

Otro material que se podría haber utilizado es el plástico, como en el caso de las impresoras 3D, pero el elevado precio de este instrumento y vistas las ventajas que proporciona la madera de contrachapado, hizo que la opción elegida fuera la madera.

Las ventajas más representativas que tiene este tipo de madera son:

Es un material **reciclable**. Para su fabricación se combinan la materia prima renovable, biodegradable y que no contamina el medio con las características de la madera.

Es un elemento **resistente**, ya que está formado por una serie de capas de madera contrapuestas entre sí, coladas y prensadas. El uso de una capa fenólica en sus caras más el tipo de fabricación que se le realiza, deriva en un material resistente y ligero.

A diferencia de otros materiales, la madera de contrachapado no se abolla ni se oxida.

➤ Forma.

La forma elegida para el robot del presente proyecto es una especie de “caja andante”, en la cual, el diseño nos permite colocar las entradas a la Raspberry Pi B+ (USB y puerto Ethernet) en la parte superior.

La placa controladora y la batería que le proporciona la alimentación necesaria irán colocadas en posición vertical en la parte trasera del robot. Por su parte, las placas con los circuitos soldados irán en la parte delantera del robot. Encima de ellas se coloca una pequeña balda que servirá como protección para las placas soldadas,



proporcionará sujeción a la Raspberry Pi B+ y servirá como apoyo para la cámara. Por último, quedará colocar la tapa de la cámara, quedando así un techo en el cual se pueden transportar cosas de bajo peso.

4.1.2 Diseño de las piezas.

En la fabricación del robot, las piezas realizadas han sido la base inferior, la balda interior, los laterales y la base superior.

➤ Base inferior.

La base inferior del robot será un rectángulo del material escogido. La pieza tendrá una longitud de 13 cm y una anchura de 11 cm. Sobre ella se colocarán las placas con los circuitos soldados junto con la Raspberry Pi B+ y la batería, estas dos últimas en posición vertical. Se le recortará un pequeño horificio para hacer hueco a la rueda de apoyo.



Figura 23. Base inferior.

➤ Base superior.

La base superior del robot tendrá varias funciones. La primera es la de tapar la cámara para que ningún agente externo al robot pueda introducirse en esa zona. La segunda función que cumple es de zona de transporte, es decir, que en este “techo” del robot se podrán colocar mercancías de poco peso para que estas sean transportadas.



Como se observa en la *figura 24*, La balda es un rectángulo de 13x11 cm con un corte en forma de “U”. Este corte es debido a la Raspberry Pi B+ y su batería, ya que irán colocadas en ese espacio.

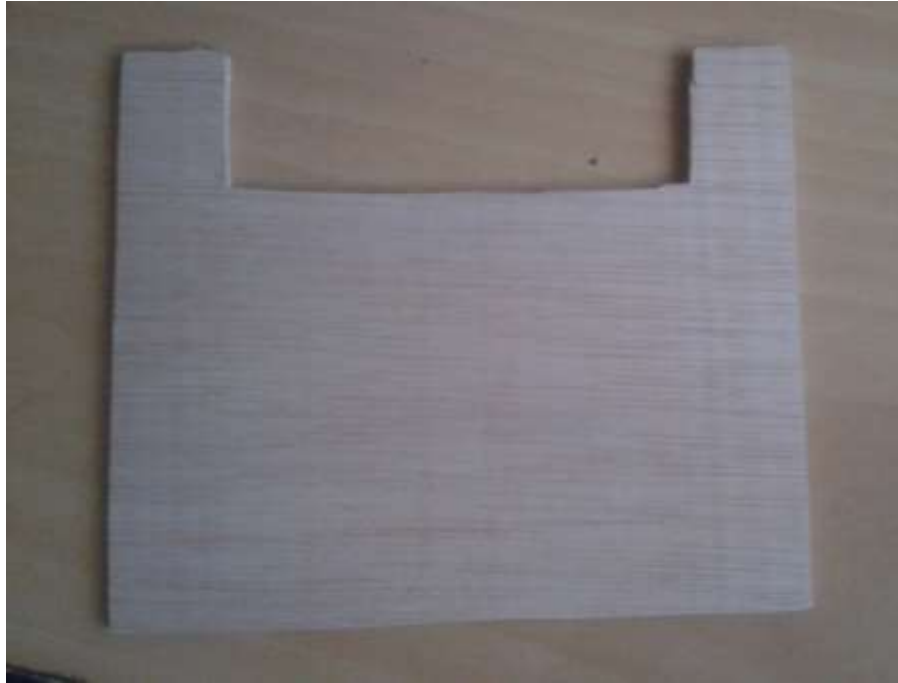


Figura 24. Balda superior o techo.

➤ **Laterales.**

Son las piezas más individualizadas del robot, ya que son 4 laterales los que lo componen y cada uno tiene una forma distinta.

Los dos laterales del robot, la parte izquierda y la derecha, son idénticos. Son dos rectángulos de madera de contrachapado con un agujero en la parte inferior que se utilizará para introducir los motores que van enganchados a las ruedas.

El lateral posterior es el más simple, puesto que su única función es la de cerrar el robot por la parte trasera del mismo.



Figura 25. Pared lateral.



La parte delantera o anterior del robot sí es algo más compleja, ya que aparte de cerrar el robot en su zona, debe de tener agujeros para la linterna compuesta de seis LED's y en la parte superior el agujero que corresponde a la cámara.

➤ **Balda interior.**

Será un rectángulo de madera, con una anchura menor que la base inferior. La diferencia será la anchura de la Raspberry Pi B+ en conjunto con la batería. Su función será doble. La primera es tratar de proteger lo máximo las placas soldadas que se situarán debajo de esta balda. La segunda, que servirá de sustento para la cámara del robot. Esta balda nos ofrece una función extra, y es que también nos proporciona una ayuda a la hora de sujetar la Raspberry para que no pierda su posición vertical.

Al igual que que la balda que actúa como techo del robot, esta balda interior es un rectángulo de 13x11 cm con un corte en forma de "U". Este corte será un poco más grande (concretamente 1 cm) que el de la balda superior, ya que por ese espacio entre la Raspberry Pi B+ y la balda deben de pasar los cables que conectan la Raspberry Pi B+ con las placas con los circuitos soldados.



Figura 26. Balda interior.



➤ **Rueda de apoyo.**

Para completar las piezas del robot necesarias, nos queda una pequeña rueda de apoyo que se le ha colocado en la parte delantera del robot, junto con un eje. La misión de esta rueda, de un diámetro bastante menor que el de las ruedas motorizadas, es la de dar un apoyo más al robot, para que la base inferior de este no roce el suelo. Así, el robot también ganará en estabilidad.



Figura 27. Rueda de apoyo.

Por lo tanto, la forma final del chasis mecánico será la siguiente (*Figura 28*).



Figura 28. Chasis mecánico del robot.

Una especie de caja con tres pisos. En el primero las placas, en el segundo la cámara y el tercero el techo del robot.

4.1.3 Fabricación y montaje.

En las siguientes tablas se enumeran las cantidades de cada tipo de pieza que se han fabricado para el robot:

➤ **Piezas construidas para el robot:**

Nombre	Cantidad
Base Superior	1
Base Inferior	1
Balda interior	1
Laterales	4

Tabla 3. Piezas construidas.

➤ **Componentes electrónicos y mecánicos necesarios:**

Nombre	Cantidad
Raspberry Pi B+	1
Cámara	1
Sensor luz (Resistencia LDR)	2
Adaptador Bluetooth	1
Mando teleoperación	1



LED blanco	6
LED verde	2
Transistor BC337	5
Transistor 2N222	2
Cables con conector	6
Resistencia 100 k Ω	4
Resistencia 1 k Ω	10
Conector	25
Motores	2
Ruedas	2
Rueda de apoyo	1

Tabla 4. Lista de componentes mecánicos y electrónicos necesarios.

➤ **Fuente de alimentación:**

Nombre	Cantidad
Batería recargable	1

Tabla 5. Lista de componentes de la fuente de alimentación

➤ **Otros componentes:**

Nombre	Cantidad
Cola para madera	1
Soldador + estaño	1
Segueta	1

Tabla 6. Lista de otros componentes.

El primer paso después de haber diseñado todas y cada una de las piezas es el de cortarlas a medida. A la vez, obtener los componentes electrónicos necesarios como también la batería y los elementos necesarios para la realización del robo, detallados en las tablas anteriores.

El primer paso es pegar la base inferior con las paredes de los motores y la pared posterior. A continuación se colocarán las placas en la base inferior y la Raspberry Pi B+, con su batería, en posición vertical, dejando los puertos USB mirando hacia arriba, así cuando finalice la fabricación del robot, estos quedan libres para poder enchufar ahí alguna mejora más del robot.

Ya con la Raspberry Pi B+, la batería, las placas y las paredes laterales y posterior colocadas, se coloca la balda interior. Sobre esta se sitúa la cámara y, a su vez, encima de esta, la base superior del robot. Por último se coloca la pared anterior para terminar de construir esta pequeña “caja andante”.

No hay que obviar, que mientras se está realizando el montaje del robot, hay que realizar los conexiones correspondientes.

La imagen final del robot, sin la pared delantera, es la siguiente:



Figura 29. Imagen final del robot.

4.2 Sistema electrónico.

Una vez finalizado el montaje mecánico del robot, procedemos a realizar el sistema electrónico.

Este sistema tratará de conectar los sensores (fotorresistencias LDR), que recopilan información sobre el entorno en el que se encuentra el robot, y la cámara, la cual transmitirá por streaming por dónde se va moviendo el robot, con el dispositivo de control (la placa Raspberry Pi B+). Esta última se encarga de manejar y organizar dicha información y ejecutar las respectivas acciones que llevarán a cabo los actuadores. También la Raspberry Pi B+ será la encargada de mantener una interacción con el mando de teleoperación por vía Bluetooth y una emisión constante de lo que ve la cámara a través de internet, gracias a una conexión Wi-Fi. Para el correcto funcionamiento, todos estos componentes deberán de estar correctamente alimentados, como veremos a continuación con los circuitos eléctricos realizados.

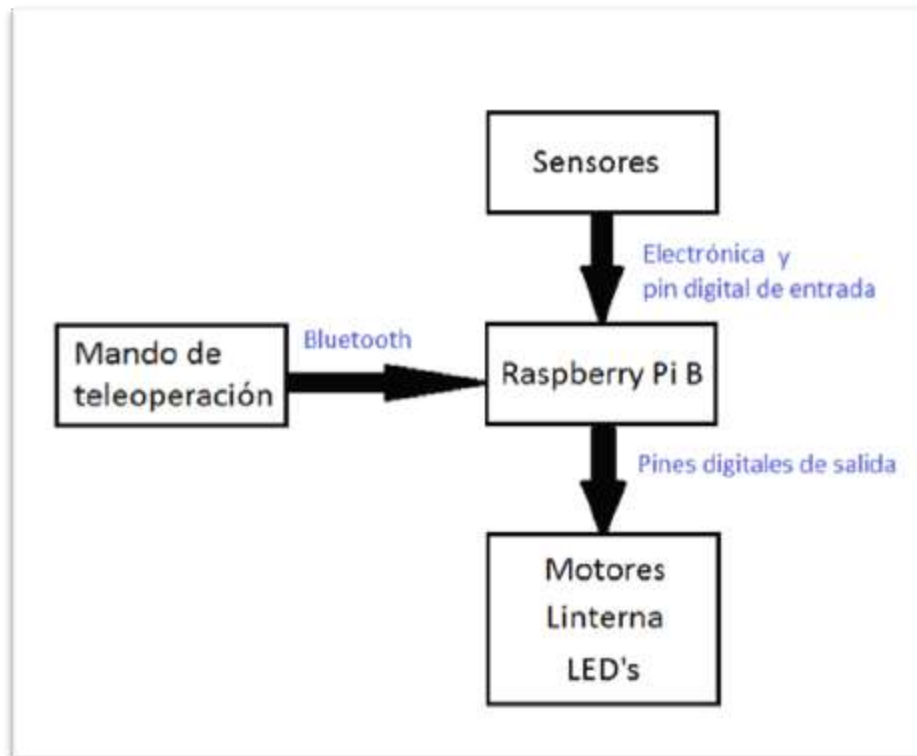


Figura 30. Esquema general del sistema electrónico.

Para la realización de dichos circuitos se han seguido tres fases en cada uno de ellos como son:

- **Estudio:** el primer paso siempre es estudiar cómo se deben de hacer las cosas.
- **Desarrollo:** Preparación y montaje de los circuitos con sus respectivas pruebas de funcionamiento.
- **Implementación:** Tras comprobar que las respectivas pruebas han tenido un correcto funcionamiento, soldar el circuito en las placas.
- **Resultados y conclusiones:** Observar si los resultados obtenidos son los deseados y, si no es así, volver al paso 1, es decir, pensar dónde está el error y corregirlo.

Para poder formar los circuitos, han sido necesarios varios tipos de componentes electrónicos:

- ✓ Resistencias de varios valores: 100k Ω y 1k Ω .
- ✓ Transistores. Dos tipos, uno el BC337 y el otro 2N2222A.
- ✓ Fotorresistencias. Dos unidades. Estos serán los sensores de luz.
- ✓ LED 's.
- ✓ Cables y conectores.
- ✓ Protoboard.
- ✓ Dos placas punto a punto.



✓ Soldador y estaño.

El robot dispondrá de un circuito para cada motor, otro para la linterna y otro para la implementación del modo automático (va hacia la luz).

Estos circuitos han sido repartidos en dos pequeñas placas, por lo que los tres primeros (ambos motores + linterna) están soldados en una primera placa y el circuito restante, el del modo automático, en otra segunda placa.

A continuación se ha realizado una explicación más individual de cada uno de los circuitos empleados.

4.2.1 Placa 1.

La primera placa estará compuesta por los dos circuitos correspondientes a cada motor y el circuito de la linterna. En este apartado se explica al detalle cada circuito y al final del mismo se muestra la imagen de la placa soldada con sus correspondientes circuitos:

➤ **Motores:**

Para la realización de este circuito, se ha necesitado un transistor BC337, un motor y los puertos correspondientes de la Raspberry Pi de alimentación de 5 V, del puerto GPIO que dé la orden de si se tiene que mover el motor o no y de tierra (GND).

Al llevar dos motores, habría que realizar dos montajes similares con un solo apunte diferente. Si para el primer motor se ha utilizado su patilla positiva a la salida del emisor y la negativa a tierra, en el segundo motor esto debe de ser al revés, porque si no, lo que provocaríamos es que cada rueda rote en sentido contrario a la otra rueda, por lo que se produciría una incongruencia en el robot, que cuando se activaran los dos motores, rotaría hacia el lado contrario al motor que más fuerza ejerza. Por eso, si en el primero la pata positiva del motor se conecta con la salida del emisor y la pata negativa a tierra, en el segundo motor la conexión debe de ser al revés, es decir, la pata positiva del motor a tierra y la negativa al emisor del transistor.

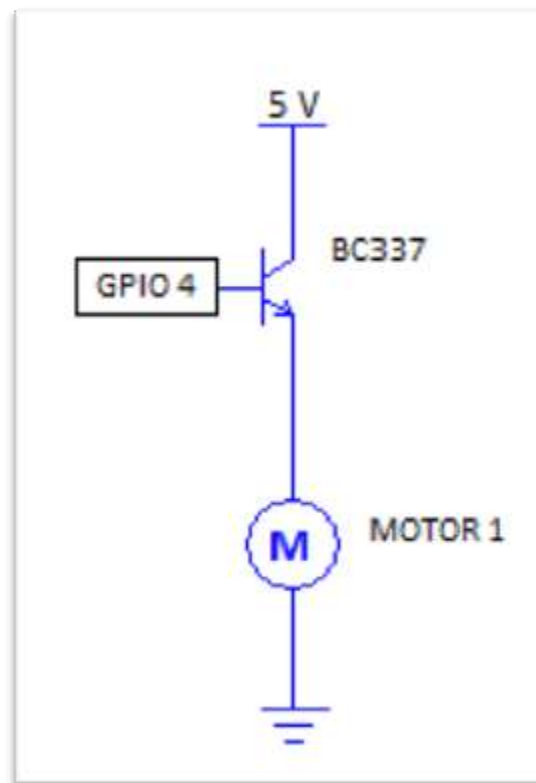


Figura 31. Circuito del motor 1.

El circuito básico para el funcionamiento del motor es el que se observa en la anterior imagen.

Dicho circuito consiste en un transistor BC337 el cual se alimenta por el colector con 5 V proporcionados por el pin 2 de la Raspberry Pi B+. En la base se conecta el puerto GPIO que se encarga de ordenar cuándo el motor tiene que moverse, que en este caso será el puerto GPIO4, que coincide con el pin 7. Para finalizar, por la patilla del emisor, se conecta a un extremo del motor. El otro extremo del motor se conecta a tierra, que también será proporcionada por la Raspberry.

De esta forma, cuando se pulse el botón correspondiente con el puerto GPIO4 y este se active, permitirá que haya una diferencia de tensión entre el colector, la base y el emisor, con el paso de corriente que esto conlleva. De esta forma al motor le llega una diferencia de tensión y una corriente que le hace mover sus engranes y permite avanzar y transportarse al robot. En caso de que el GPIO 4 esté desactivado porque su botón no ha sido pulsado, no circulará corriente a través del transistor y el motor permanecerá quieto.



Cabe decir, que este circuito es para la utilización de un motor y el robot tendrá dos, por lo que para el segundo motor se deberá de implementar un circuito igual al explicado en este apartado.

Para el segundo motor, se utilizará otro puerto de la Raspberry Pi, y será el GPIO 17 que corresponde al pin 11. El funcionamiento obviamente será el mismo.

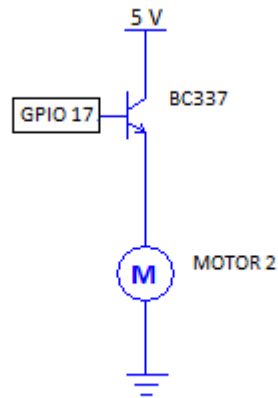


Figura 32. Circuito del motor 2.

➤ **Linterna:**

El circuito que permite activar los seis LED 's que forman la linterna es sencillo.

Para la realización de este circuito se utiliza las conexiones con los pines correspondientes de la Raspberry Pi B+ de la alimentación de 5 V y de tierra, a parte de los seis LED 's con sus respectivas resistencias.

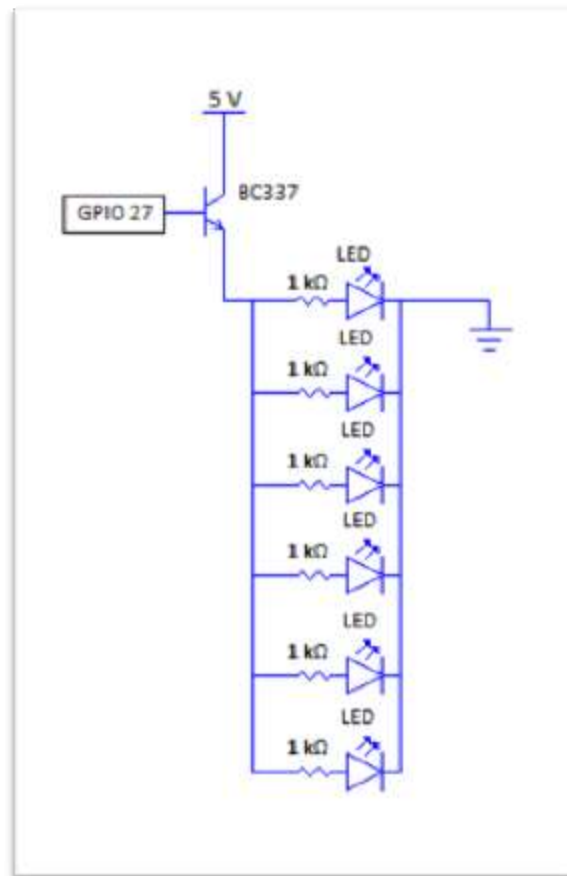


Figura 33. Circuito de la linterna.

Como se observa en la imagen del circuito, éste consta de una alimentación de 5 V que proporciona la Raspberry Pi B+ en su pin 2. Esta tensión se conectará a una resistencia, que a su vez, se unirá a la patilla positiva de uno de los LED 's. La siguiente resistencia con su LED, se situarán en paralelo con el anterior LED y su resistencia, y así sucesivamente con los cuatro LED 's restantes, hasta el último. Por lo tanto, los seis LED 's contarán con su pata positiva conectada a la resistencia, que a su vez está conectada a la alimentación, y su pata negativa conectada a tierra, proporcionada de nuevo por la Raspberry Pi B+ y uno de sus 8 posibles pines de tierra o GND.

El voltaje proporcionado por la Raspberry Pi B+ llegará al circuito a través de un transistor BC337, que actúa como si fuera un interruptor. Cuando el puerto GPIO27 este activado, llegará voltaje a la base del transistor, lo que derivará que este actúe como un interruptor cerrado. En caso de que el BOTÓN + no sea pulsado, el GPIO27 estará desactivado, haciendo así que no llegue voltaje a la base del BC337. De este modo, el transistor actúa como un interruptor abierto, haciendo que no llegue potencial a las resistencias ni a los LED 'S, por lo que, hasta que el GPIO27 no sea activado, no se encenderá los LED 's (la linterna).



En el caso de que llegue tensión al emisor del transistor, la corriente circulará por todos y cada uno de los LED 's, lo que hará que estos se enciendan. Al estar conectados en paralelo, sobre los seis LED 's caerá la misma tensión y, al ser el mismo dispositivo con características idénticas o casi iguales, la intensidad será la misma (o muy parecida) para todos los LED 's.

En las siguientes imágenes se puede observar el esquema de la placa para su soldadura y el resultado final:

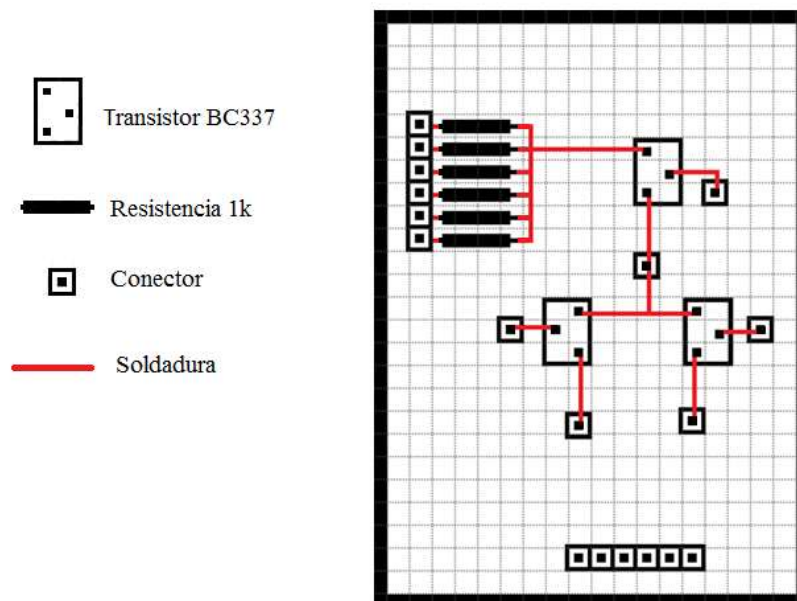


Figura 34. Esquema de la placa 1.

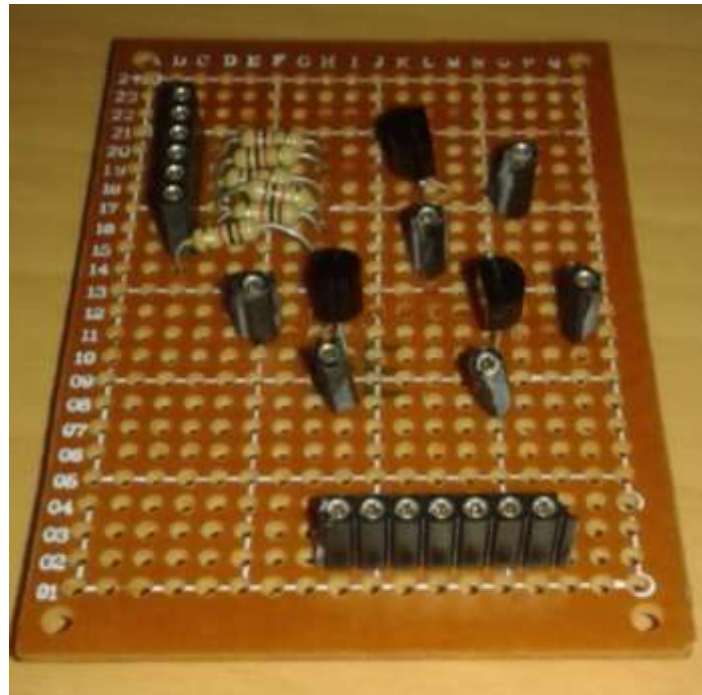


Figura 35. Imagen final de la placa 1.

4.2.2 Placa 2.

En esta segunda placa se ha implementado el circuito del modo automático, ya que es el más complejo y el que más espacio ocupa.

➤ Modo automático.

Este circuito es el primer paso para llegar al circuito final, el cual tendrá más complejidad debido a las modificaciones que ha sufrido este primer circuito inicial, para conseguir los objetivos marcados para este trabajo. Para su realización se debe disponer de:

- 1 transistor 2N2222A.
- 3 resistencias (1 de 100k Ω y otras dos de 1k Ω).
- 1 fotorresistor (LDR).
- 1 LED.
- Cables y una placa punto a punto.

El primer circuito que se realiza es el siguiente:

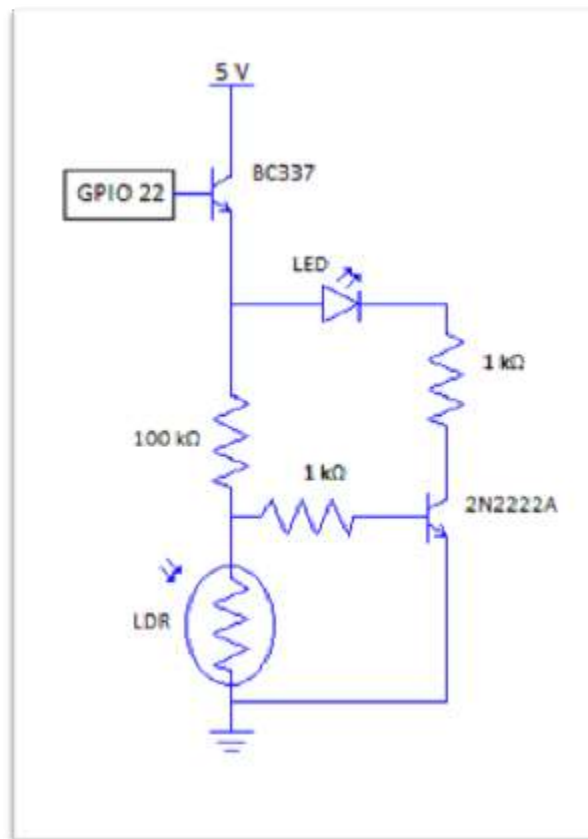


Figura 36. Circuito inicial del modo automático.

Cuando el modo automático es activado, el robot irá hacia la luz sin ninguna falta de que sea teleoperado.

La fuente de alimentación de este primer circuito, viene dada por un transistor, con los 5 V proporcionados por uno de los pines de la Raspberry Pi y el puerto GPIO 22, correspondiente al pin 15. A esta alimentación se le conecta en paralelo una resistencia de 100 kΩ y un LED.

a) Tras la resistencia, mediante una conexión en paralelo, se sitúa el fotorresistor, que se conectará a tierra, y otra resistencia, esta vez de 1 kΩ y que se conecta con la base del transistor.

b) El LED se coloca a la salida de la alimentación. A su pata negativa, se le conecta una nueva resistencia de 1 kΩ que se une al colector del transistor.

Ya en el transistor, sólo falta por unir la pata del emisor a la tierra.



A continuación se procede a explicar cómo funciona eléctricamente el circuito:

Lo primero a analizar es el sub-circuito formado por el transistor BC337, la alimentación de 5 V que emite la Raspberry y el puerto GPIO 22. Este transistor actúa de una forma semejante a un interruptor. Y es que dependiendo del GPIO 22, permitirá que por el emisor circule la corriente proveniente de la fuente, o no. El entendimiento es sencillo: si el GPIO 22 está desactivado, provoca que a la base del transistor no llegue voltaje, por lo que el BC337 actuaría como un interruptor abierto. Esto se traduce en que el LED, capte luz o no el sensor (la fotorresistencia LDR), no lucirá porque nunca le llega un voltaje a su pata positiva. Por el contrario, si el puerto GPIO 22 está activado, deriva en una descarga de 3.3 V en la base. Esto a su vez provoca que el transistor actúe como interruptor cerrado y de esta forma se alimenta al circuito principal.

Este circuito detecta la ausencia de luz, es decir, que el LED lucirá cuando el sensor no detecte luz. El funcionamiento de este circuito se basa en la fotorresistencia. Y es que dependiendo de la cantidad de luz que esta reciba, la corriente pasará por ella o se irá por la otra parte del circuito. Por lo tanto, si al sensor le llega mucha luz, el valor de la resistencia del fotorresistor bajará y pasará a través de él la corriente, dejando a la base del transistor sin tensión, por lo que el transistor actuaría como un interruptor abierto. Al LED, le llegaría por su pata positiva la tensión dada por la resistencia que va conectada al colector y por la pata negativa llegaría al transistor, que, al actuar como circuito abierto, sería el culpable de que el LED no luzca. Si por el contrario, el fotorresistor no detecta luz, el valor resistivo del mismo aumentará, de tal forma que crea una gran diferencia de tensión entre sus extremos. Por ello, a la base del transistor le llegaría cierto voltaje, que permitiría que actuase como un interruptor cerrado entre su colector y su emisor, ocasionando el encendido del LED.

El segundo paso del desarrollo de este circuito es conseguir el efecto inverso del LED, es decir, que ahora el LED luzca cuando al sensor le llegue luz, ya que lo que se busca con este circuito es que el robot reaccione ante la luz, no ante su ausencia.

Para su realización se debe de poseer los mismos componentes que en el primer circuito.

El conexionado es muy parecido al visto anteriormente, solamente se ha de cambiar de posición al LED y situarlo en el colector del transistor, después de la resistencia de 1 k Ω . Después, se deberá de conectar a tierra el LED y la resistencia a la alimentación.



Su funcionamiento es semejante al explicado en el apartado anterior. Ambos transistores en este caso actúan como un interruptor. Si a la base le llega un voltaje, se asemeja a un interruptor cerrado entre el colector y el emisor. Si por el contrario no le llega tensión a la base, actuaría parecido a un interruptor abierto entre colector y emisor.

El circuito es el siguiente:

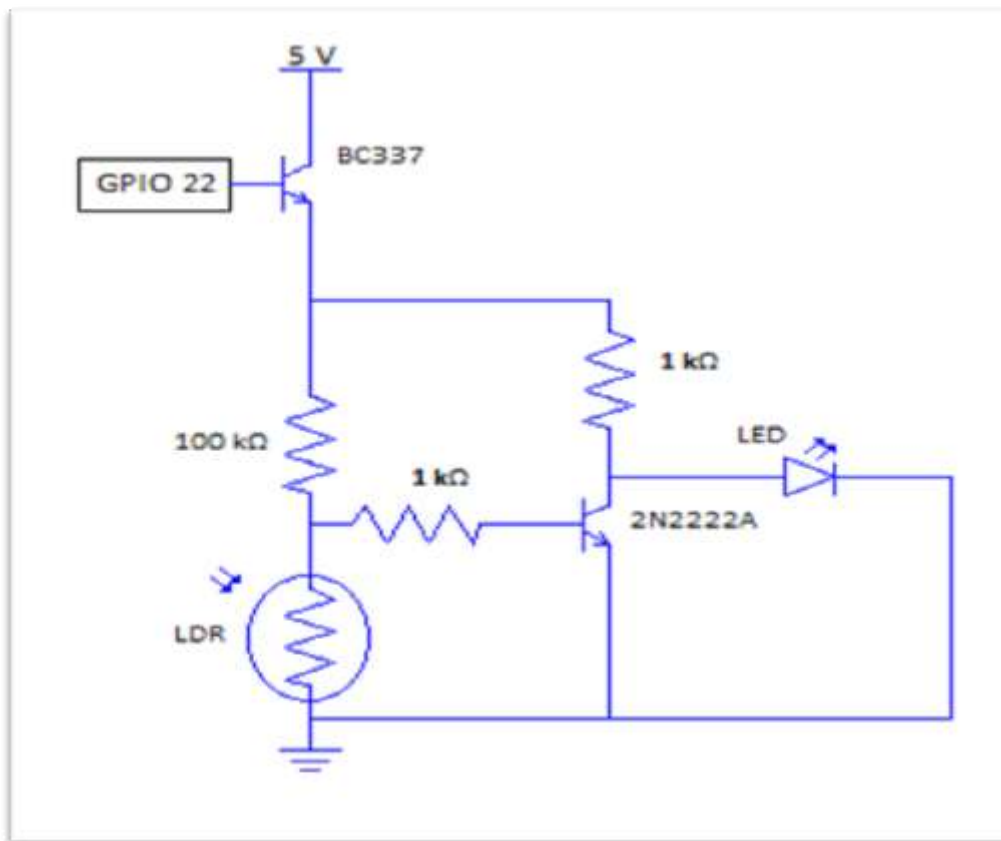


Figura 37. Modificación del circuito inicial del modo automático.



En este circuito se observa que, cuando el sensor LDR detecta mucha luz, su valor resistivo baja, propiciando que la corriente circule a través de él y haciendo que se conecte la base a la tierra. Por lo tanto, de este modo, no llegaría ninguna tensión a la base del transistor, que actuaría como un interruptor abierto. Así, lo que se ve desde el LED, es que en una pata le llega la alimentación, por la parte de arriba del circuito, y la otra pata está conectada a tierra, por lo que se crea una diferencia de potencial sobre el LED que hace que luzca.

En caso contrario, es decir, que al sensor no le llegue luz, su valor resistivo aumenta, creando una gran tensión entre sus extremos. De este modo, a la base del transistor le llegará una tensión, lo que provoca que éste actúe como un interruptor abierto. Así, desde el LED, ya no se ve que le llegue alimentación, si no que ahora ve que están sus dos patas conectadas a tierra, por lo que no se crea una diferencia de potencial sobre él, que deriva en que no luzca.

Por último sólo queda incluir la Raspberry en el circuito, para que ella pueda manejar la información que proporcionan los sensores. Este es el último paso y será el circuito final.

Para la realización de dicho conexionado se necesita:

- 3 transistores: 1 transistor 2N2222A y 2 BC337.
- Raspberry Pi B+.
- 1 LED.
- 1 fotorresistencia LDR.
- 3 resistencias: una de 100 k Ω y otras dos de 1 k Ω .
- Cables y una placa punto a punto.

Las cantidades de los componentes citados son para la fabricación de un circuito, que sirve para un motor. En el caso de este proyecto, el robot lleva dos motores por lo que habría que multiplicar por dos cada componente.

El montaje del circuito es el siguiente:

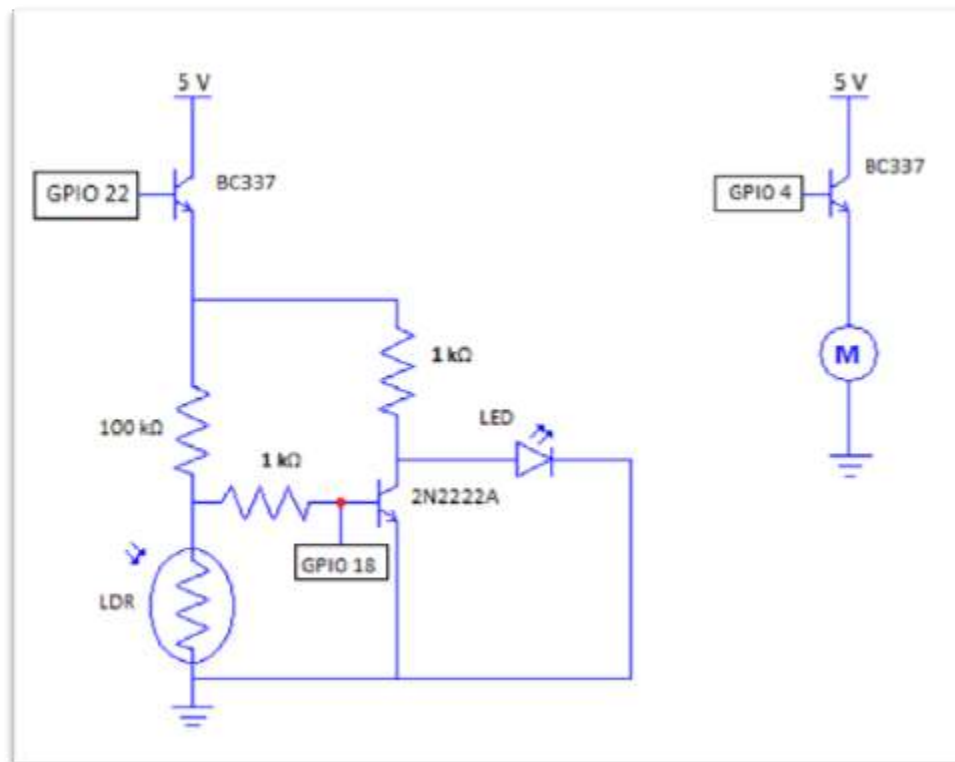


Figura 38. Circuito final del modo automático (para el motor 1).

El funcionamiento de este circuito se basa en el circuito del paso anterior. La única modificación que sufre es que se introducen los puertos GPIO18, GPIO 4 y GPIO 11 en el circuito. De nuevo, el voltaje necesario para que el circuito sensible a la luz funcione, lo proporcionará uno de los puertos de la Raspberry Pi, concretamente el GPIO 22 (pin 15).

Como vemos, depende de nuevo de un transistor. Como se ha explicado en el apartado anterior, si a la base del transistor le llega corriente, este actuará como un interruptor cerrado entre colector y emisor. Si, por el contrario, a la base no le llega tensión, actuará como un interruptor abierto.

El otro puerto GPIO que se conecta es el GPIO18 (pin 12). Este puerto actúa como entrada, detectando si al LED le llega o no tensión. En caso de que le llegue voltaje, mediante el programa creado, se activará el GPIO7 y el GPIO11 que son los que tienen el control de los motores. Así, se activará el puerto correspondiente al motor del lado contrario del LED que detecte la luz. Por ejemplo, si el LED que detecta luz es el que está situado a la derecha, se moverá el motor situado en la parte izquierda para así girar a la derecha, que es de donde proviene la luz.



La linterna, cuando se active el modo automático se apagará.

Este circuito se implementaría para un motor. Para que el segundo motor también funcione respecto a la luz que le llega al sensor, habría que hacer el mismo circuito, solo cambiaría el puerto GPIO utilizado como entrada, que para el primer motor es el GPIO 18 y para el segundo sería el GPIO 23.

Su circuito sería el siguiente:

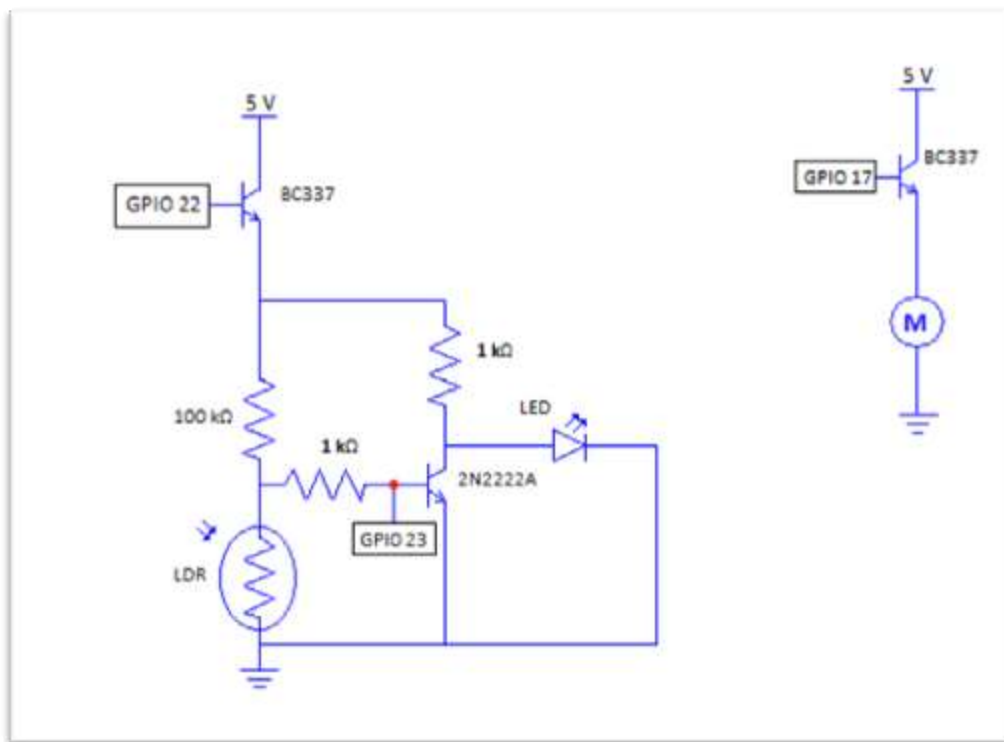


Figura 39. Circuito final del modo automático (para el motor 2).

En las siguientes imágenes se puede observar los esquemas del circuito para su posterior soldadura y el resultado final:

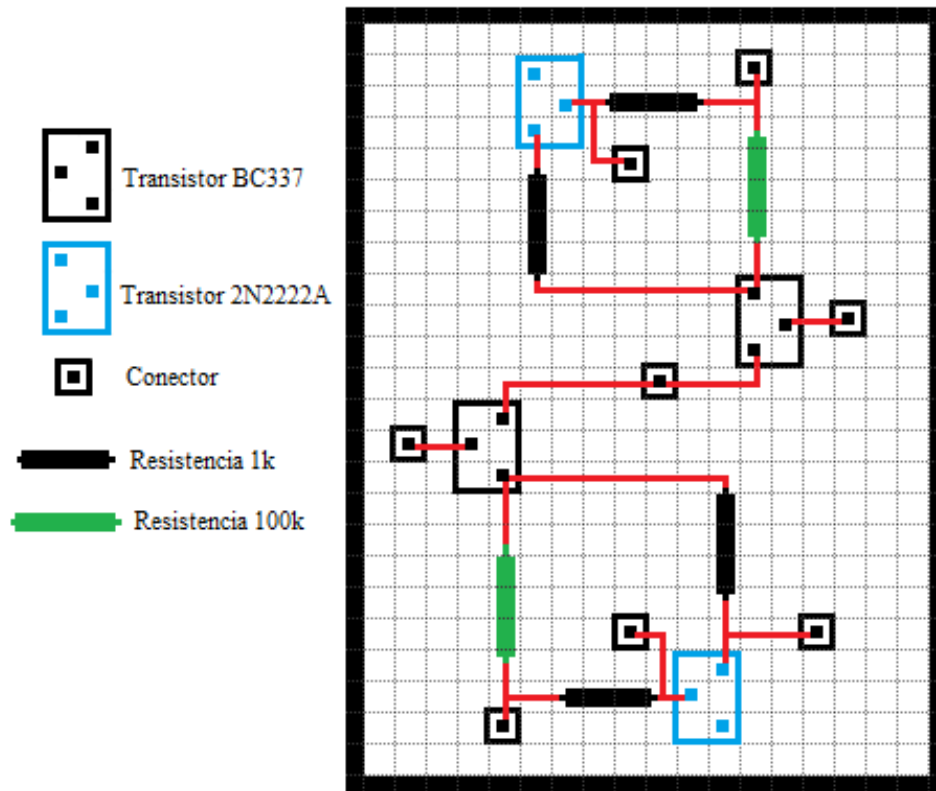


Figura 40. Esquema de la placa 2.



Figura 41. Imagen final de la placa 2.



4.3 Programación.

Una vez realizados el sistema mecánico y el sistema electrónico con sus correspondientes conexiones, se procede a configurar todos los componentes para el control del robot. En esta sección se va a explicar los códigos implementados para cada acción realizable por parte del robot.

Los códigos están realizados en lenguaje Python. Python es un lenguaje de programación desarrollado como proyecto de código y cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.

4.3.1 Descarga del Sistema Operativo.

Lo primero que se realiza, es la descarga de un sistema operativo para la Raspberry Pi B+. En el caso del presente proyecto es Raspbian, pero hay otros muchos sistemas operativos disponibles para descargarse.

El archivo descargado es una imagen con el sistema operativo correspondiente, lo único que queda es introducir dicho archivo en una tarjeta micro-SD y ésta última, introducirla en la ranura correspondiente de la Raspberry Pi B+.

Para ello será necesario el programa Win32DiskImagr. Es un programa simple de usar en el cual sólo hace falta señalar el archivo que se quiere introducir en la tarjeta micro-SD.

Cuando la imagen se ha creado correctamente en la tarjeta micro-SD, el programa nos avisará y se podrá retirar la tarjeta micro-SD del ordenador para proceder a introducirla en la Raspberry Pi B+.

Ya con la tarjeta introducida en la Raspberry Pi B+, se puede conectar esta última a la luz y a un monitor o una televisión, a través de un cable HDMI. Para acabar este apartado, lo único que queda es realizar una configuración personal inicial.

4.3.2 Instalaciones necesarias.

Tras acabar el último apartado, hace falta una serie de instalaciones para la cámara y para el mando de teleoperación. Estas descargas se realizarán desde el terminal, pero para ello, lógicamente, es necesario que la Raspberry Pi B+ esté conectada a Internet, ya sea por medio de cable o inalámbricamente.

➤ Cámara:

Para poder visualizar lo que la cámara está “viendo”, es necesario seguir unas pautas para poder descargar el software *motion*. Una de las virtudes que tiene este software,



es que permite distinguir el área en el que se ha producido movimiento. Lo hace recuadrando la zona en la que se ha producido dicho movimiento.

Para su descarga, se debe de escribir una serie de sentencias en la consola del software Raspbian, las cuales permite que se actualice la Raspberry Pi B+, comprobar si la cámara está conectada o descargar e instalar el software motion, que es imprescindible. Las sentencias son las siguientes:

1. Actualizar la Raspberry Pi:

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

2. Comprobar si la cámara está conectada.

2.1. Listar los dispositivos de USB.

```
lsusb
```

2.2. Ver el ID (sustituyendo el ID de la Raspberry en la X)

```
lsusb -s1:X -v
```

3. Instalar el V4L.

```
sudo apt-get install v4l-utils
```

4. Confirmar que hay un dispositivo de video conectado.

```
ls -al /dev/video*
```

5. Mostrar información de la cámara.

```
sudo v4l2-compliance -d /dev/video0
```



6. Instalar motion.

```
sudo apt-get install motion
```

7. Configurar motion.

7.1. Abrir el primer fichero de configuración.

```
sudo nano /etc/motion/motion.conf
```

7.1.1. Cambiar los siguientes parámetros:

- daemon OFF → ON.
- webcam_localhost ON → OFF.
- Locate OFF → ON.
- Width 320 → 640.

- Height 240 → 480.

7.2. Abrir el segundo fichero de configuración:

```
sudo nano /etc/default/motion
```

7.2.1. Cambiar el siguiente parámetro:

- start_motion_daemon NO → YES.

7.3. Iniciar la emisión del video.

- **Cambiando la configuración:** `sudo service motion start`
- **Sin cambiar la configuración:** `Sudo service motion restar`

Después de todo esto, ya se podrá ver por streaming qué emite la cámara. Solo queda abrir el navegador Mozilla Firefox, e introducir en la búsqueda los números de la IP de la Raspberry Pi B+ utilizada y el puerto, por ejemplo: 192.168.0.17:8081.

Con el único navegador que funciona es con Mozilla Firefox.



➤ **Mando de teleoperación:**

El mando de teleoperación será un mando de la consola Wii, de la compañía Nintendo.

Al igual que para la cámara, para poder dar una buena utilidad al mando hace falta un software, el cual reconozca qué botón se está pulsando y qué tiene que hacer en ese caso la Raspberry Pi B+ y, a su vez, el robot.

A continuación se explicarán los pasos a dar para la instalación de este software:

1. Actualizar la Raspberry Pi:

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

2. Instalar los drivers de Bluetooth, ya que este tipo de conexión será la vía de comunicación entre el mando y la placa.

```
sudo apt-get install --no-install-recommends bluetooth
```

3. Instalar el paquete python-cdwiid.

```
sudo apt-get install python-cwiid
```

4. En este momento, se introduce en uno de los puertos USB de la Raspberry Pi el adaptador Bluetooth.

4.1. Comprobación de que el servicio de bluetooth funciona correctamente.

```
sudo /etc/init.d/bluetooth status
```

4.2. Si se recibe un mensaje como este: [ok] bluetooth is running. es que el servicio está funcionando. Si no es así, probar a reiniciar la Raspberry Pi.



4.3. Presionar los botones 1 y 2 del mando Wii. Los LED 's azules empiezan a parpadear y escribir en la consola:

`hcitool scan`

4.4. Si todo el proceso va bien, saldrá un mensaje como el siguiente:

`Scanning ... 00:19:1D:8F:BA:F7 Nintendo RVL-CNT-01`

Esta es la direccion mac del mando reconocido por la Raspberry Pi.

5. Instalación de wminput

`sudo apt-get install wminput`

6. Los ficheros de configuración se encuentran en /etc/cwiid/wminpuut y vienen varios ejemplos. Para probar las distintas configuraciones hay que escribir el siguiente comando:

`wminput -c <nombre_fichero_configuracion> -d 00:19:1D:8F:BA:F7`

Uno de estos ficheros de ejemplo ha sido programado exclusivamente para este proyecto. Así el robot hará finalmente lo que el usuario le indique con el mando.

4.3.3 Código.

Como se puede observar en el apartado anterior, en el paso 5, con la sentencia “sudo apt-get install wminput”, nos descargamos varios ficheros de wminput. Uno de ellos es el fichero del mando de la consola Wii. Ese código inicial es el siguiente:

```
# This program utilises the cwiiid Python library in order to get input
over bluetooth from a wiimote.
# The following lines of code demonstrate many of the features realted
to wiimotes, such as capturing button presses and rumbling the
controller.
# I have managed to map the home button to the accelerometer - simply
hold it and values will appear!

# Coded by The Raspberry Pi Guy. Work based on some of Matt Hawkins's!

import cwiiid, time
```



```
button_delay = 0.1

print 'Please press buttons 1 + 2 on your Wiimote now ...'
time.sleep(1)

# This code attempts to connect to your Wiimote and if it fails the
program quits
try:
    wii=cwiid.Wiimote()
except RuntimeError:
    print "Cannot connect to your Wiimote. Run again and make sure you
are holding buttons 1 + 2!"
    quit()

print 'Wiimote connection established!\n'
print 'Go ahead and press some buttons\n'
print 'Press PLUS and MINUS together to disconnect and quit.\n'

time.sleep(3)

wii.rpt_mode = cwiid.RPT_BTN

while True:

    buttons = wii.state['buttons']

    # Detects whether + and - are held down and if they are it quits the
program
    if (buttons & cwiid.BTN_PLUS & cwiid.BTN_MINUS == 0):
        print '\nClosing connection ...'
        # NOTE: This is how you RUMBLE the Wiimote
        wii.rumble = 1
        time.sleep(1)
        wii.rumble = 0
        exit(wii)

    # The following code detects whether any of the Wiimotes buttons
have been pressed and then prints a statement to the screen!
    if (buttons & cwiid.BTN_LEFT):
        print 'Left pressed'
        time.sleep(button_delay)

    if (buttons & cwiid.BTN_RIGHT):
        print 'Right pressed'
        time.sleep(button_delay)

    if (buttons & cwiid.BTN_UP):
        print 'Up pressed'
        time.sleep(button_delay)

    if (buttons & cwiid.BTN_DOWN):
        print 'Down pressed'
        time.sleep(button_delay)

    if (buttons & cwiid.BTN_1):
        print 'Button 1 pressed'
        time.sleep(button_delay)

    if (buttons & cwiid.BTN_2):
        print 'Button 2 pressed'
```



```
time.sleep(button_delay)

if (buttons & cwiid.BTN_A):
    print 'Button A pressed'
    time.sleep(button_delay)

if (buttons & cwiid.BTN_B):
    print 'Button B pressed'
    time.sleep(button_delay)

if (buttons & cwiid.BTN_HOME):
    wii.rpt_mode = cwiid.RPT_BTN | cwiid.RPT_ACC
    check = 0
    while check == 0:
        print(wii.state['acc'])
        time.sleep(0.01)
        check = (buttons & cwiid.BTN_HOME)
    time.sleep(button_delay)

if (buttons & cwiid.BTN_MINUS):
    print 'Minus Button pressed'
    time.sleep(button_delay)

if (buttons & cwiid.BTN_PLUS):
    print 'Plus Button pressed'
    time.sleep(button_delay)
```

Tabla 7. Código inicial.

Es un código sencillo que lo primero que pide es que el BOTÓN 1 y BOTÓN 2 sean pulsados simultáneamente para que se produzca la conexión. En caso contrario, la conexión mando-Raspberry Pi B+ se perderá.

Posteriormente, cuando se pulsa un botón, el nombre de este aparecerá por pantalla.

Por último, para desconectar el mando, se debe de pulsar simultáneamente el BOTÓN + y el BOTÓN -.

El código descargado contiene comentarios en inglés.

A continuación se expone cómo debe de quedar el código final para que el robot siga las instrucciones descritas en este documento.

➤ **Cabecera:**

Esta parte del código, se dedica para declarar las librerías utilizadas y las variables globales que, en caso de este programa y de este proyecto, serán los puertos GPIO correspondientes de la Raspberry Pi.



Se declaran las librerías *cwiid*, las librerías para poder utilizar los puertos GPIO y la librería del reloj.

Posteriormente, se declara el modo GPIO en BOARD, es decir, que los números utilizados para referirnos a los puertos GPIO en el programa serán los números correspondientes a sus pines.

```
# Conectar la Raspberry Pi al mando Wii via bluetooth y leer los
estados de los botones en python.

# Importar las librerias de Python requeridas

import cwiid
import RPi.GPIO as GPIO
import time

button_delay = 0.1

GPIO.setmode(GPIO.BOARD)

# Se declara los puertos GPIO 7,11,13,15 como SALIDAS.
GPIO.setup(7,GPIO.OUT)
GPIO.setup(11,GPIO.OUT)
GPIO.setup(13,GPIO.OUT)
GPIO.setup(15,GPIO.OUT)
# Se declara el puerto GPIO 12 como ENTRADA.
GPIO.setup(12,GPIO.IN)
GPIO.setup(23,GPIO.IN)
```

Tabla 8. Cabecera.

➤ **Conexión:**

Lo primero que realizará el programa será la comprobación para saber si la conexión se ha hecho debidamente. Esta conexión se hace pulsando el BOTÓN 1 y BOTÓN 2 simultáneamente, después de que el código haya sido lanzado por el terminal de la Raspberry Pi B+. Si la conexión es satisfactoria, saldrá un mensaje en el terminal de la Raspberry Pi B+ y se podrá ratificar notando una pequeña vibración en el mando. Si por el contrario la conexión ha sido errónea, saldrá otro mensaje diciendo que no se ha podido establecer comunicación entre el mando y la Raspberry Pi B+.

```
# Instruccion para encender el mando
print '\n\n\n\n*****'
print '*   Presione los botones 1 y 2 a la vez   *'
print '*****'
time.sleep(1)

# Conecta el mando Wii. Si ocurre algun error avisa de que no se ha
podido establecer la conexion.
# Si no se produce la conexion se sale del programa.
```



```
try:
    wii=cwiid.Wiimote()
except RuntimeError:
    print '...ERROR...\n'
    print 'No se ha podido establecer la conexion.'
    quit()
```

Tabla 9. Comprobación de conexión mando-Raspberry Pi B+.

El siguiente paso que da el programa es saber si son pulsados los botones BOTÓN + y BOTÓN - simultáneamente, para saber si debe de salirse de la aplicación.

```
# En caso de que la conexion sea satisfactoria.
wii.rumble = 1
time.sleep(1)
wii.rumble = 0
print 'Mando teleoperador conectado correctamente.\n'
print 'Presione el BOTON + y el BOTON - para salir del programa y
cerrar la conexion.\n'

wii.rpt_mode = cwiid.RPT_BTN

while True:

    buttons = wii.state['buttons']

    # Si se pulsa el BOTON + y el BOTON - a la vez, se cierra la
    conexion.
    # together then rumble and quit.
    if (buttons - cwiid.BTN_PLUS - cwiid.BTN_MINUS == 0):
        print '\nCerrando la conexion ...'
        wii.rumble = 1
        time.sleep(1)
        wii.rumble = 0
        print 'Conexion cerrada satisfactoriamente.'
    exit(wii)
```

Tabla 10. Comprobación para desconexión.

Lo restante es programar los que realizará cada botón cuando sea pulsado.

El código para teleoperar los motores se basa en 4 botones: BOTÓN 2, BOTÓN UP, BOTÓN DOWN y BOTÓN 1.

BOTÓN 2: Este botón sirve para activar ambos motores. La programación se basa en la estructura *if*. Si se pulsa dicho botón, los puertos correspondientes a los dos motores (GPIO7 y GPIO11) se activarán.



```
#BOTON 2.  
#ACELERAR. Se encienden ambos motores.  
if (buttons & cwiid.BTN_2):  
    print 'Button 2 pressed'  
    time.sleep(button_delay)  
    GPIO.output(7,GPIO.HIGH)  
    GPIO.output(11,GPIO.HIGH)  
    GPIO.output(15,GPIO.LOW)
```

Tabla 11. Código BOTÓN 2.

BOTÓN UP: Su programación se inicia con un *else* referido al *if* del anterior botón. Es decir, que si el BOTÓN 2 no ha sido pulsado, y pulsas el BOTÓN UP, el robot girará a la izquierda y, para eso, debe de funcionar el motor derecho y pararse el izquierdo. Así, esta parte del código consta de un GPIO que se desactiva y de otro que se activa.

```
else:  
    #BOTON UP.  
    #GIRAR HACIA LA IZQUIERDA. Se enciende el motor derecho  
    solamente.  
    if (buttons & cwiid.BTN_UP):  
        print 'Up pressed'  
        time.sleep(button_delay)  
        GPIO.output(7,GPIO.HIGH)  
        GPIO.output(11,GPIO.LOW)  
        GPIO.output(15,GPIO.LOW)
```

Tabla 12. Código BOTÓN UP.

BOTÓN DOWN: Al igual que el BOTON UP, esta porción de código empieza con el *else* referido al *if* del anterior botón explicado. Por lo tanto, si no han sido pulsados ni BOTÓN 2 ni BOTÓN UP, si pulsas BOTÓN DOWN, el robot para girar a su lado derecho, moverá el motor contrario al movido si pulsas el BOTÓN UP. Por lo tanto, el GPIO que antes se desactivaba, ahora se activa. Lo contrario ocurre con el puerto que antes se quedaba encendido, que ahora se apaga.

```
else:  
    #BOTON DOWN.  
    #GIRAR HACIA LA DERECHA. Se enciende el motor izquierdo  
    solamente.  
    if (buttons & cwiid.BTN_DOWN):  
        print 'Down pressed'  
        time.sleep(button_delay)  
        GPIO.output(7,GPIO.LOW)  
        GPIO.output(11,GPIO.HIGH)  
        GPIO.output(15,GPIO.LOW)
```

Tabla 13. Código BOTÓN DOWN.



Existe una parte de código debajo del BOTÓN DOWN, que sirve para ver si se está pulsando algún botón. En caso de que ningún botón se pulse, los motores estarán quietos. Si se borra esta parte de código, lo único que pasaría es que para mover al robot se seguirían utilizando los botones ya explicados, pero al soltarlos no frenaría solo, sino que habría que pulsar el BOTÓN 1. Un ejemplo. Si se pulsa el botón dos durante 2 segundos y se suelta, cuando pasen esos 2 segundos, los motores se detendrán. Sin la parte del código mencionada, pasados esos dos segundos, ninguno de los motores no parará, hasta que pulsemos el BOTÓN 1.

```
else:
    #Si ningun boton es pulsado, los motores estan los dos
    apagados.
    GPIO.output(7, GPIO.LOW)
    GPIO.output(11,GPIO.LOW)
```

Tabla 14. Código ausencia de botón pulsado.

BOTÓN 1: Este botón sirve para desactivar ambos motores. Su programación es sencilla como se ve en la tabla.

```
#BOTON 1.
#PARAR. Desactiva ambos motores.
if (buttons & cwiid.BTN_1):
    print 'Button 1 pressed'
    time.sleep(button_delay)
    GPIO.output(7,GPIO.LOW)
    GPIO.output(11,GPIO.LOW)
```

Tabla 15. Código BOTÓN 1.

➤ **Linterna:**

El código para activar la linterna también es sencillo. Se basa en activar o desactivar un puerto GPIO. El puerto utilizado para esta función es el GPIO27 que corresponde en la Raspberry Pi al pin 13.

La activación de la linterna se ha implementado en el BOTÓN +.

```
#BOTON +.
#ACTIVAR LA LINTERNA. Activa el GPIO correspondiente.
if (buttons & cwiid.BTN_PLUS):
    print 'Plus Button pressed'
    time.sleep(button_delay)
    GPIO.output(13, GPIO.HIGH)
```

Tabla 16. Código BOTÓN +.

La desactivación de la linterna se ha implementado en el BOTÓN -.



```
#BOTON -.
#DESACTIVAR LA LINTERNA. Desactiva el GPIO correspondiente.
if (buttons & cwiid.BTN_MINUS):
    print 'Minus Button pressed'
    time.sleep(button_delay)
GPIO.output(13,GPIO.LOW)
```

Tabla 17. Código BOTÓN -.

➤ **Modo automático:**

Esta parte del código es la más compleja. Se basa en un par de *if* 's.

El primero es para saber si pulsas el BOTÓN A, como los *if* 's de cualquier otro botón. En ese caso, lo primero que hará el programa será apagar ambos motores, apagar la linterna y encender el GPIO22, que es el que deja pasar la alimentación al circuito detector de luz.

Después de esto, al tener conectado el GPIO18, correspondiente al pin 12, al LED para saber si por el circula corriente o no, sabremos si se debe de encender el motor de ese circuito. Con un *if* comprobamos si al LED le llega voltaje. Si es que no, el GPIO 12 pasará a valer 1, por lo que entraríamos en la estructura del citado *if* y, dependiendo de si lo detectan ambos sensores o uno de ellos, el robot se moverá hacia delante o girará para un lado. Si por el contrario al LED no le llega tensión, querrá decir que el GPIO18 seguirá valiendo 0, por lo que el programa pasará por el *else* del *if* anterior y ambos motores estarán apagados, ya que a sus sensores correspondientes no les ha llegado luz.

```
#BOTON A.
#ACTIVAR MODO AUTOMATICO. Si un sensor detecta luz, se enciende el
motor correspondiente.
if (buttons & cwiid.BTN_A):
    print 'Button A pressed'
    time.sleep(button_delay)
    GPIO.output(13, GPIO.LOW)
    GPIO.output(15, GPIO.HIGH)
    if GPIO.input(12)==0:                #si el puerto GPIO 18 detecta
un voltaje
        GPIO.output(7, 1)                #activa el motor
correspondiente.
    else:
        GPIO.output(7, 0)

        if GPIO.input()==0:              #Lo mismo que el anterior if
pero con el segundo motor.
            GPIO.output(7, 1)
        else:
            GPIO.output(7,0)
```

Tabla 18. Código BOTÓN A.



El BOTÓN B tendrá la función de desconectar el modo automático. Lo hará cortando la salida de voltaje por el GPIO 22. También desactivará los GPIO 's correspondientes a ambos motores (GPIO 4 y GPIO 17). El puerto correspondiente a la linterna, el GPIO 27, también será desactivado. Así, pulsar el BOTÓN B será prácticamente como reiniciar la aplicación y empezar de cero.

```
#BOTON B.  
#DESACTIVAR MODO AUTOMATICO. Desactiva el modo automatico, apaga los  
motores y la linterna hasta nueva orden.  
if (buttons & cwiid.BTN_B):  
    print 'Button B pressed'  
    time.sleep(button_delay)  
    GPIO.output(7, GPIO.LOW)  
    GPIO.output(11, GPIO.LOW)  
    GPIO.output(13, GPIO.LOW)  
    GPIO.output(15, GPIO.LOW)
```

Tabla 19. Código BOTÓN B.

4.3.4 Funciones del mando de teleoperación.

En esta sección se va a explicar con más detalle las funciones que puede activar cada botón del mando de teleoperación.

Cada botón que se pulse del mando, mandará una orden a la Raspberry Pi B+ mediante la conexión Bluetooth. Ésta a su vez, realizará las acciones correspondientes dependiendo de la orden que le haya llegado a la placa controladora. La siguiente tabla explica la acción que se lleva a cabo con cada botón:



BOTÓN	FUNCIÓN	ESTADOS
UP	Hace rotar al robot hacia su izquierda	GPIO4: ON GPIO17: OFF
RIGHT	Sin función	-----
LEFT	Sin función	-----
DOWN	Hace rotar al robot hacia su derecha	GPIO4: OFF GPIO17: ON
A	Activa el modo automático (robot va hacia la luz)	Se activa el motor del lado contrario en la que se detecta la luz.
B	Desactiva el modo automático	Vuelve a depender de la teleoperación.
+	Activa la linterna	GPIO27: ON
HOME	Sin función	-----
-	Desactiva la linterna	GPIO27: OFF
1	Sin función	
2	Acelera el robot(mover hacia adelante)	GPIO4: ON GPIO 17: ON

Tabla 20. Funciones del mando de teleoperación.

Para una mejor comprensión de esta tabla y de los códigos vistos anteriormente, se muestra a continuación los diagramas de flujo de los tres circuitos del proyecto (motores, linterna y modo automático):



➤ Motores

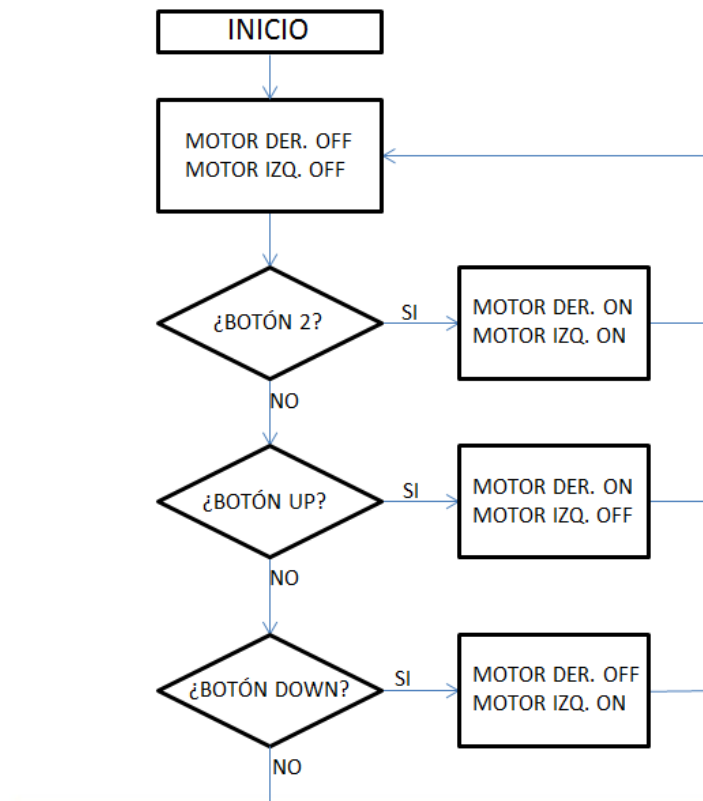


Figura 42. Diagrama de flujo de ambos motores.



➤ Linterna

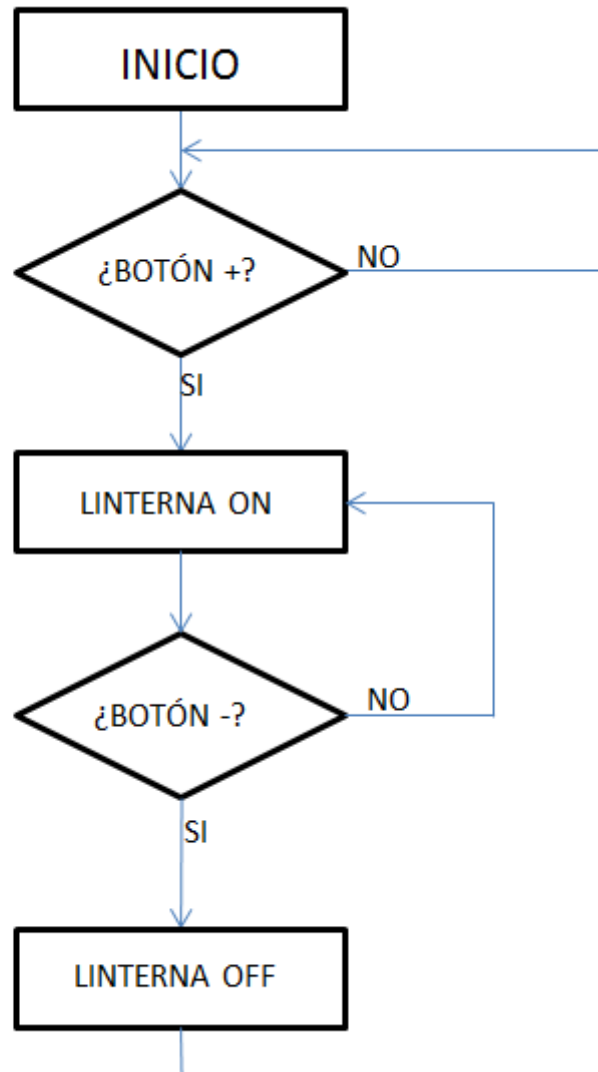


Figura 43. Diagrama de flujo de la linterna.



➤ **Modo automático**

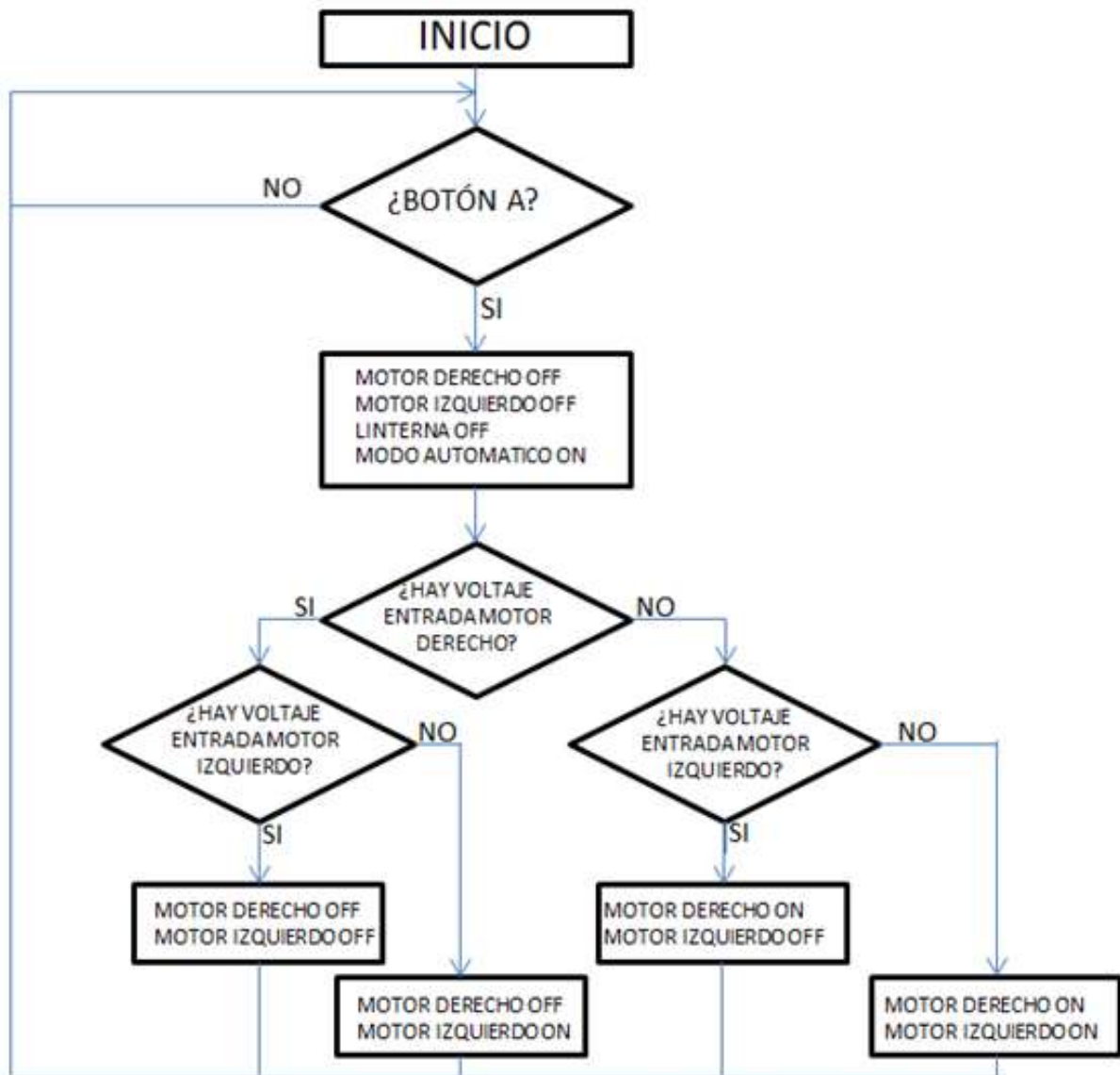


Figura 44. Diagrama de flujo del modo automático.



Capítulo 5: Pruebas y cambios.

En este capítulo se habla sobre las pruebas y los cambios que se han realizado tras la fabricación del robot.

Las fases de pruebas se dividirán en el orden de los capítulos anteriores: mecánica, electrónica y programación.



5.1 Prueba del sistema mecánico.

En este apartado se explicarán las pruebas y los cambios realizados en torno al sistema mecánico del robot.

➤ Material:

En un principio se optó por construir el robot en una caja de plástico duro. La poca manejabilidad del material hizo que se cambiara por plástico blando, que era más fácil de utilizar en la fase de construcción, pero era muy poco consistente. La idea final fue realizar el chasis del robot en un material algo más duro y resistente, perdiendo la ventaja de no ser tan ligero pero sin perder la ventaja de ser manejable. También se gana en consistencia, dureza y resistencia.

➤ Ruedas:

La primera idea era la de realizar la construcción del robot con dos ruedas, sin necesitar una pequeña rueda de apoyo.

Al término de una primera construcción de prueba, el robot volcaba hacia delante y arrastraba la parte delantera de la base inferior por el suelo, por lo que la medida a tomar era clara: colocar una rueda de apoyo. Se cambió el modelo original, que era poner las ruedas a la mitad de la distancia que mide el largo de las paredes laterales, por poner las ruedas a 2 cm de la pared posterior y formar un triciclo con la rueda delantera, colocando esta última centrada y más próxima a la parte delantera del robot.

5.2 Prueba del sistema electrónico.

En este capítulo se expondrán los resultados obtenidos tras la fabricación del robot.

Los resultados experimentales se dividen en los distintos circuitos que han sido elaborados. Unos son de mayor complejidad que otros, por lo que han entrañado más dificultad para que funcionase como se deseaba.

Los experimentos han sido desarrollados en distintos entornos.

Los circuitos diseñados para este proyecto fueron realizados en una placa protoboard. En esta placa fueron probados con un correcto funcionamiento. A continuación se realizaron las soldaduras correspondientes en ambas placas y se volvieron a probar los circuitos, para comprobar si todo había quedado soldado como debía y el robot



respondería de la forma correcta. En este apartado no hubo ninguna incidencia reseñable, puesto que todo funcionó correctamente.

➤ **Circuitos impresos:**

- Linterna: El circuito de la linterna es sencillo, al igual que ha sido sencilla su implementación. Cuando el GPIO 27 se activa, la linterna también. Si por el contrario, el puerto GPIO 27 está desactivado, la linterna no lucirá.
- Motores: Circuito muy parecido al de la linterna. Sencillo de implementar y de fácil entendimiento. Si se pulsa el botón correspondiente, se enciende el motor. La complejidad en este caso surgió a la hora de programar. A la hora de la preparación del robot surgió un problema, y es que ambos motores se conectaron del mismo modo: pata positiva al positivo y la pata negativa a tierra. Lo que pasó fue que a la hora de montar el robot las ruedas giraban en sentido opuesto, por lo que hubo que cambiar de polaridad uno de los motores (es indiferente).
- Modo automático: De nuevo en este apartado, la distancia, mientras se esté dentro del rango permitido por el adaptador de Bluetooth conectado a la Raspberry Pi B+(15 metros), no es el problema.

El circuito se fue realizando paso a paso, dando algún que otro problema pero teniendo cada uno su inmediata solución.

La luz que el sensor debía de captar era la emitida por un flash de un teléfono móvil.

Primero fue la electrónica. Los resultados del primer experimento que se hizo fueron buenos, ya que el circuito hacía lo que se pedía que en ese momento era hacer lucir el LED cuando el sensor no captara luz. El siguiente paso fue implementar el circuito para que el LED actuara totalmente al revés, y se consiguió. Aquí el mayor problema que surgió fue la captación de luz. En entornos totalmente oscuros lo capta bastante mejor que en entornos medios, como por ejemplo estar cerca de la ventana y que el sensor capte un poco de luz.



La distancia también es un problema, ya que al LED solo le llegaba voltaje como para encenderse cuando el flash lucía cerca del sensor, como a menos de 1 metro de distancia. Esto puede ser debido a la baja calidad de la fotorresistencia LDR.

➤ **Batería:**

Para que el robot no sea esclavo de una conexión por cable, la solución es colocar una batería o un portapilas. La conclusión fue colocar una batería con una conexión de salida compatible con la entrada de alimentación de la Raspberry Pi B+. Este tipo de conexión es micro-USB y se eligió una batería con salida de 5 V a 2.000 mA, es decir, las mismas características que tiene el cable que conecta la Raspberry Pi B+ a la luz.

5.3 Pruebas de programación.

Las pruebas realizadas en esta sección dieron buenos resultados, ya que todos los archivos que se iban compilando, daban el rendimiento correcto.

Por su parte, con la instalación del software Raspbian en la Raspberry Pi B+ no surgió ningún problema inesperado. Tampoco surgió ningún contratiempo con la descarga de los archivos *motion*, que sirven para la emisión por streaming de la cámara a través de la Raspberry Pi B+, ni con los archivos descargados para conectar el mando de teleoperación a la placa controladora.



Capítulo 6: Conclusión.

En este capítulo se desarrollarán las conclusiones sobre el proyecto.

Se explicarán también futuros trabajos que se puedan implementar en el robot para complementar su rendimiento actual.



6.1 Conclusiones sobre el proyecto.

Como conclusión tras la realización de todas las pruebas realizadas, se puede concluir que el presente proyecto que ha consistido en el diseño, la fabricación y el control de un pequeño robot móvil teleoperado ha cumplido con los objetivos propuestos.

Se ha realizado el diseño de las piezas que componen el robot, como también se han elegido y adquirido la totalidad de los componentes electrónicos que han sido instalados en el proyecto. Se ha programado la teleoperación del robot y también, cuando el usuario lo desee, el modo automático, el cual hará que el robot se mueva hacia la luz. También se puede seguir los pasos del robot a través de un aparato electrónico con conexión a Internet y con el navegador *Mozilla Firefox*, gracias al software instalado para la cámara del robot.

Como conclusión final sobre el robot, se puede decir que se ha creado un robot eficaz, asequible y abierto a futuros trabajos.

- ✓ **Eficaz** -> con escasos recursos técnicos y económicos se ha logrado fabricar un robot totalmente funcional.
- ✓ **Asequible** -> cualquier persona sería capaz de lograr su construcción.
- ✓ **Abierto a futuros trabajos** -> es un trabajo que se deja puertas abiertas de cara al futuro, ya que se puede seguir ampliando sus funciones y complementándolo más.

La conclusión personal que puedo sacar de este proyecto es que me ha ayudado mucho a ampliar mis conocimientos en el campo de la robótica. También me ha ayudado a intentar superarme, ya que me he tenido que enfrentar a diversos y complejos problemas sobre la realización de este proyecto, consiguiendo superar cada uno de ellos.

- Entendimiento de cómo funcionan los transistores y los demás componentes electrónicos utilizados.
- Aprendizaje del lenguaje Python a nivel básico.
- Práctica del manejo de los puertos GPIO con sus entradas y salidas.
- Experiencia del uso de una placa como la Raspberry Pi, un dispositivo capaz de ofrecer muchas posibilidades y campos de trabajos.



6.2 Trabajos futuros.

Este proyecto, como ya se ha explicado en el apartado anterior, es un trabajo abierto a futuras mejoras.

Una de ellas es simplemente incorporarle más sensores. Un ejemplo sería incorporarle un sensor de ultrasonidos para evitar chocar con las paredes en el modo automático.

Otra mejora podría ser un diseñar, fabricar y programar un modo rastreo, en el que el robot, conteniendo un receptor de infrarrojos, busque un emisor.

El manejo a través de voz sería otra importante mejora, pudiendo así transformarse en un robot educativo o, incluso, en un robot que ayude a personas ancianas o discapacitadas.

Otra mejora a añadir podría ser el añadirle una ventosa, un cepillo o un pequeño objeto que pueda ser lanzado, con la finalidad de ser tratado de una forma más infantil, como un juguete.



Capítulo 7: Presupuesto.

A continuación se desarrollará el presupuesto general para desarrollar el presente proyecto. En él se incluirán todos los gastos, desde cada uno de los componentes utilizados en el robot hasta las horas de trabajo dedicadas al proyecto.



7.1 Presupuesto general.

CÓDIGO	DESCRIPCIÓN	CANTIDAD	PRECIO UNITARIO	PRECIO TOTAL
01	BLOQUE 1: SOFTWARE			0,00 €
01.01	Win32DiskImager	1	0	0 €
01.02	Raspbian	1	0	0 €
02	BLOQUE 2: COMPONENTES ELECTRÓNICOS			7,60 €
02.01	Fotorresistencia LDR	2	0,50	1 €
02.02	Resistencia 100 k Ω	4	0,05	0,20 €
02.03	Resistencia 1 k Ω	6	0,05	0,30 €
02.04	Transistor BC337	5	0,30	1,50 €
02.05	Transistor 2N222	2	0,70	1,40 €
02.06	LED blanco	6	0,30	1,80 €
02.07	LED verde	2	0,30	0,60 €
02.08	Regleta pines	1	0,80	0,80 €
03	BLOQUE 3: COMPONENTES MECÁNICOS			17,00 €
03.01	Motor	2	5	10,00 €
03.02	Rueda	2	2,50	5,00 €
03.03	Madera	1	2,00	2,00 €
04	BLOQUE 4: TECNOLOGÍA INALÁMBRICA			10,00 €
04.01	Adaptador Bluetooth	1	10,00	10,00 €
04.02	Mando	1	0,00	0,00 €
05	BLOQUE 5: MICROCONTROLADOR			40,00 €
05.01	Raspberry Pi B+	1	35,00	35,00 €
05.02	Carcasa Raspberry Pi B+	1	5,00	5,00 €
06	BLOQUE 6: ALIMENTACIÓN			5,00 €
06.01	Batería	1	5,00	5,00 €
07	BLOQUE 7: OTROS COMPONENTES			0,00 €
07.01	Cámara	1	0,00	0,00 €
07.02	Cola, Soldador y estaño	1	0,00	0,00 €
TOTAL PROYECTO				79,60 €

Tabla 21. Presupuesto general.

Como se puede observar la tabla del presupuesto, el bloque de mayor precio es el del microcontrolador, con una diferencia notable sobre el resto. El segundo bloque más caro es el mecánico, pero sin llevarse mucha diferencia de precio con los bloques 2, 4 y 6. Respecto a los bloques 1 y 7, salen a coste cero, es decir, gratis, debido a que la



obtención de esos productos no ha supuesto más gasto, ya sea por su fácil obtención o porque ya se poseían antes de empezar el presente proyecto.

A esto habría que añadirle las horas de trabajo en el proyecto. El precio del proyecto ascendería a unos 4.000 €, ya que son aproximadamente 4 meses de trabajo.

7.2 Fechas de desarrollo.

Fase 1. Planificación

- Estudio del lenguaje Python.
🕒 15 días.
- Estudio de la Raspberry Pi B+.
🕒 5 días

Fase 2. Desarrollo

- Descarga y programación del fichero utilizado.
🕒 15 días.
- Análisis y diseño de circuitos.
🕒 30 días.
- Análisis y diseño del chasis.
🕒 10 días.
- Fabricación e implementación.
🕒 15 días.
- Evaluación de resultados.
🕒 5 días.

Fase 3. Documentación.

- Memoria Trabajo Fin de Grado.
🕒 30 días.
- Presentación audiovisual.
🕒 5 días.



Capítulo 8: Anexos.

En este último capítulo podremos ver los anexos, con el código al completo utilizado, como la bibliografía.



8.1 Código completo.

```
# Conectar la Raspberry Pi al mando Wii via bluetooth y leer los
estados de los botones en python.

# Importar las librerias de Python requeridas

import cwiid
import RPi.GPIO as GPIO
import time

button_delay = 0.1

GPIO.setmode(GPIO.BOARD)

# Se declara los puertos GPIO 7,11,13,15 como SALIDAS.
GPIO.setup(7,GPIO.OUT)
GPIO.setup(11,GPIO.OUT)
GPIO.setup(13,GPIO.OUT)
GPIO.setup(15,GPIO.OUT)
# Se declara el puerto GPIO 12 como ENTRADA.
GPIO.setup(12,GPIO.IN)

# Instruccion para encender el mando
print '\n\n\n\n*****'
print '*   Presione los botones 1 y 2 a la vez   *'
print '*****'
time.sleep(1)

# Conecta el mando Wii. Si ocurre algun error avisa de que no se ha
podido establecer la conexion.
# Si no se produce la conexion se sale del programa.
try:
    wii=cwiid.Wiimote()
except RuntimeError:
    print '...ERROR...\n'
    print 'No se ha podido establecer la conexion.'
    quit()

# En caso de que la conexion sea satisfactoria.
wii.rumble = 1
time.sleep(1)
wii.rumble = 0
print 'Mando teleoperador conectado correctamente.\n'
print 'Presione el BOTON + y el BOTON - para salir del programa y
cerrar la conexion.\n'

wii.rpt_mode = cwiid.RPT_BTN

while True:

    buttons = wii.state['buttons']

    # Si se pulsa el BOTON + y el BOTON - a la vez, se cierra la
conexion.
    # together then rumble and quit.
    if (buttons - cwiid.BTN_PLUS - cwiid.BTN_MINUS == 0):
        print '\nCerrando la conexion ...'
```



```
wii.rumble = 1
time.sleep(1)
wii.rumble = 0
print 'Conexion cerrada satisfactoriamente.'
exit(wii)

# Comprueba si se pulsan otros botones bit a bit. Cada boton tiene
una funcionalidad.

#BOTON 1.
#PARAR. Desactiva ambos motores.
if (buttons & cwiid.BTN_1):
    print 'Button 1 pressed'
    time.sleep(button_delay)
    GPIO.output(7,GPIO.LOW)
    GPIO.output(11,GPIO.LOW)

#BOTON -.
#DESACTIVAR LA LINTERNA. Desactiva el GPIO correspondiente.
if (buttons & cwiid.BTN_MINUS):
    print 'Minus Button pressed'
    time.sleep(button_delay)
    GPIO.output(13,GPIO.LOW)

#BOTON +.
#ACTIVAR LA LINTERNA. Activa el GPIO correspondiente.
if (buttons & cwiid.BTN_PLUS):
    print 'Plus Button pressed'
    time.sleep(button_delay)
    GPIO.output(13, GPIO.HIGH)

#BOTON A.
#ACTIVAR MODO AUTOMATICO. Si un sensor detecta luz, se enciende el
motor correspondiente.
if (buttons & cwiid.BTN_A):
    print 'Button A pressed'
    time.sleep(button_delay)
    GPIO.output(13, GPIO.LOW)
    GPIO.output(15, GPIO.HIGH)
    if GPIO.input(12)==0:          #si el puerto GPIO 18 detecta un
voltage
        GPIO.output(7, 1)      #activa el motor correspondiente.
    else:
        GPIO.output(7, 0)

        if GPIO.input()==0:          #Lo mismo que el anterior if
pero con el segundo motor.
            GPIO.output(7, 1)
        else:
            GPIO.output(7, 0)

#BOTON B.
#DESACTIVAR MODO AUTOMATICO. Desactiva el modo automatico, apaga los
motores y la linterna hasta nueva orden.
if (buttons & cwiid.BTN_B):
    print 'Button B pressed'
    time.sleep(button_delay)
    GPIO.output(7, GPIO.LOW)
```



```
GPIO.output(11, GPIO.LOW)
GPIO.output(13, GPIO.LOW)
GPIO.output(15, GPIO.LOW)

#BOTON LEFT.
#NINGUNA FUNCIONALIDAD.
if(buttons & cwiid.BTN_LEFT):
    print 'Left pressed'
    time.sleep(button_delay)

#BOTON RIGHT.
#NINGUNA FUNCIONALIDAD.
if(buttons & cwiid.BTN_RIGHT):
    print 'Right pressed'
    time.sleep(button_delay)

#BOTON 2.
#ACELERAR. Se encienden ambos motores.
if (buttons & cwiid.BTN_2):
    print 'Button 2 pressed'
    time.sleep(button_delay)
    GPIO.output(7,GPIO.HIGH)
    GPIO.output(11,GPIO.HIGH)
    GPIO.output(15,GPIO.LOW)

else:
    #BOTON UP.
    #GIRAR HACIA LA IZQUIERDA. Se enciende el motor derecho
    solamente.
    if (buttons & cwiid.BTN_UP):
        print 'Up pressed'
        time.sleep(button_delay)
        GPIO.output(7,GPIO.HIGH)
        GPIO.output(11,GPIO.LOW)
        GPIO.output(15,GPIO.LOW)

    else:
        #BOTON DOWN.
        #GIRAR HACIA LA DERECHA. Se enciende el motor izquierdo
        solamente.
        if (buttons & cwiid.BTN_DOWN):
            print 'Down pressed'
            time.sleep(button_delay)
            GPIO.output(7,GPIO.LOW)
            GPIO.output(11,GPIO.HIGH)
            GPIO.output(15,GPIO.LOW)

        else:
            #Si ningun boton es pulsado, los motores estan los dos
            apagados.
            GPIO.output(7, GPIO.LOW)
            GPIO.output(11,GPIO.LOW)

#BOTON HOME.
#NINGUNA FUNCIONALIDAD.
if (buttons & cwiid.BTN_HOME):
    print 'Home Button pressed'
```



```
time.sleep(button_delay)
```

Tabla 22. Código completo.

8.2 Glosario.

cm: centímetros.

GPIO: Entrada/Salida de Propósito General (General Purpose Input Output).

LDR: Light Dependant Resistor.

LED: Light-emitting diode.

mA: mili-amperios.

N.A.S.A.: Administración Nacional de la Aeronáutica y del Espacio (National Aeronautics and Space Administration).

RAM: memoria de acceso aleatorio (Random Access Memory).

TFG: Trabajo Fin de Grado.

UC3M: Universidad Carlos III de Madrid.

USB: Bus Universal en Serie (Universal Serial Bus).

V: Voltios.

€: Euros.



8.3 Bibliografía.

- [1] A. Barrientos, Fundamentos de Robótica, S.A. MCGRAW-HILL, 2007.
- [2] <http://blog-italia.com/actualidad/curiosidades/el-leon-mecanico-de-leonardo-un-curioso-automatismo.html> [Última visita:15/11/2015].
- [3] <https://es.wikipedia.org/wiki/Cibern%C3%A9tica> [Última visita:15/11/2015].
- [4] <http://proton.ucting.udg.mx/materias/robotica/> [Última visita:21/11/2015].
- [5] <http://dle.rae.es/?id=WYRIhzm> [Última visita:21/11/2015].
- [6] https://www.nasa.gov/mission_pages/msl/index.html [Última visita:10/12/2015].
- [7] Francisco Rodríguez Díaz/Manuel Berenguel Soria, Control y robótica en agricultura, UNIVERSIDAD DE ALMERIA, 2004.
- [8] <https://books.google.es/books?id=JPgyRgn-j1YC&pg=PA179&dq=robot+con+ruedas&hl=es&sa=X&ved=0ahUKEwi0vJvDg4PLAhXMPxoKHT7TB-QQ6AEIMDAA#v=onepage&q=robot%20con%20ruedas&f=false> [Última visita:12/12/2015].
- [9] <https://es.wikipedia.org/wiki/Roomba> [Última visita:12/12/2015].
- [10] http://e-archivo.uc3m.es/bitstream/handle/10016/16682/PFC_Santiago_Alfaro_Ballesteros.pdf?sequence=1 [Última visita:12/12/2015].
- [11] <http://www.raspberrystore.es/> [Última visita:21/12/2015].
- [12] <https://www.unocero.com/2012/08/02/raspbian-sistema-operativo-gratuito-para-la-raspberry-pi/> [Última visita:21/12/2015].
- [13] <http://ingeniatic.euitt.upm.es/index.php/tecnologias/item/500-lrdfotorresistencia> [Última visita:21/12/2015].
- [14] <https://es.wikipedia.org/wiki/Bluetooth> [Última visita:22/12/2015].
- [15] <http://www.electronicaembajadores.com/Productos/Detalle/-1/MM00125/motor-miniatura-gran-potencia-con-reductora-6v---298-1> [Última visita:27/12/2015].